UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Master Thesis

Master's Degree in Research and Innovation
Information and Communications Technologies
(i2-ICT)

Hierarchical text clustering applied to taxonomy evaluation

STUDENT: **Samuel Muñoz Hidalgo**
TUTOR: **David Camacho Fernández**

2014

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Máster Universitario en Investigación e Innovación en TIC (i2-TIC)

Trabajo Fin Máster

Hierarchical text clustering applied to taxonomy evaluation

Autor: Samuel Muñoz Hidalgo

Director: David Camacho Fernández

TRIBUNAL:

David Camacho Fernández (U.A.M.)

Iván Cantador Gutiérrez (U.A.M.)

David Fernández Barrero (U.A.H.)

CALIFICACIÓN: .....

FECHA: .....

Dedicado ti

Porque aprecias el trabajo
y la excelencia
donde otros se aburren
entre mares de letras.
Porque buscas el saber
y el buen hacer
compensándolo siempre bien.

Porque sabes que algo
podría estar mejor.
Pero a lo hecho,
el hecho,
no quita valor.

Pero, sobre todo,
porque descubriste
al sentirte solo
ese dia triste,
que el mundo nada vale
sin poder parar, disfrutar
y llenar el pecho de aire.

One machine can do the work of fifty ordinary men.
No machine can do the work of one extraordinary man.

Elbert Hubbard - A Thousand and One Epigrams.

In the margin for error lies all our room for maneuver.

James Geary - My Aphorisms.

ALL ANIMALS ARE EQUAL
BUT SOME ANIMALS ARE MORE EQUAL THAN OTHERS.

George Orwell - Animal Farm.

Los curas y taberneros
son de la misma opinión,
cuantos más bautizos hacen
más pesetas al cajón,
los curas y taberneros.

Que si que, que no que - Jota leonesa.

Al tío Tomasón,
le gusta el perejil
en invierno y en abril,
más con la condición,
perejil don, don,
perejil don, don,
la condición,
que llene el perejil
la boca de un lechón.

Morito pititón - Canción popular de Burgos

Contents

Summary

In computer science, the use for taxonomies is widely embraced in fields such as Artifial Inteligence, Information Retrieval, Natural Language Processing or Machine Learning. This concept classifications provide knowledge structures to guide algorithms on the task to find an acceptable-to-nearly-optimal solution on non deterministic problems.
The main problem with taxonomies is the huge amount of effort that requires to build one. Traditionally, this is done by human means and involves a team of experts to assure the quality of the result. Since this is evidently the way to get the best taxonomy possible (knowledge is an exclusive quality of humans), due to the manpower factor, it seems to be neither the fastest nor the cheapest one.
This thesis makes an extensive review of the state of the art on taxonomy induction techniques as well as ontology evaluation methods. It claims the need for a fast, automatic and arbitrary-domain taxonomy generation method and justifies the chose of the Wikipedia encyclopedia as the dataset. A framework to deal with taxonomies is proposed and implemented. In the experiments chapter, two statements are successfully refuted: the Wikipedia categorization system forms an acyclic directed graph, and the longest path between two nodes is equivalent to the taxonomic organization. Finally the framework is used to explore three arbitrary domains.

Keywords: Taxonomy Induction, Ontology Learning, Natural Languaje Processing, Wikipedia, Data Driven, Knowledge Organization, Text Clustering, Data Analysys.

Chapter 1

Introduction

Taxonomy

1. The branch of science concerned with classification, especially of organisms; systematics [28].
2. A system for naming and organizing things, especially plants and animals, into groups that share similar qualities [6].
3. Taxonomy is the practice and science of classification [46].

1.1 Motivation

In computer science, the use for taxonomies is widely embraced in fields such as Artifial Inteligence, Information Retrieval, Natural Language Processing, Machine Learning, etc. This concept classifications provide knowledge structures to guide algorithms on the task to find an acceptable-to-nearly-optimal solution on non deterministic problems. Therefore, a method to obtain a qualitative **taxonomy** over an arbitrary domain quickly, will be of great use. This generation method shall be performed mostly, **automatically**.

1.2 Identified problems

The main problem with taxonomies is the huge **amount of effort** that requires to build one. Traditionally, this is done by human means and involves a team of experts to assure the quality of the result. Since this is evidently the way to get the best taxonomy possible (knowledge is an exclusive quality of humans), due to the manpower factor, it seems to be neither the fastest nor the cheapest one. Because of the existence of many tasks where a fast result prevails over a better but slower one: data exploration/visualization, overview of the dataset, prototyping, investigation; automatization needs arise.

Like many investigation projects, finding a suitable **dataset** that allows testing both alternatives and methods is an annoyance. But, fortunatelly, the Wikipedia encyclopedia seems to be the perfect choice as it provides a corpus of documents as pages and a structure over them as a net of categories.

And last but not least, there is a need to **evaluate** the results and give a measure of the generated taxonomies.

1.3 Goals

The aim of this thesis is to investigate in automatic taxonomy generation methods and compare their performance, build a final taxonomy and measure its quality. A set of tools will be coded leading to a framework that will allow further investigations as well as other non-scientific applications related with the field.

At the end of the project, most of the following goals should be achieved.

- Investigate on taxonomy generation methods.
- Investigate on taxonomy evaluation.
- Investigate on data knowledge visualization and propose a tool.
- Investigate how clustering methods can improve the quality of taxonomies.
- Retrieve Wikipedia pages and relationships efficiently.
- Build Wikipedia taxonomies over arbitrary domains.
- Automate the proccess as much as possible.
- Publish the work.
  - This Master Thesis.
  - Release the tools as an open source library.
  - Publish a paper on a relevant medium.

1.4 Steps to take

This are the step to take in order to fulfil the goals.

1. Generate an extense revision of the state of the art.
2. Get the suitable tools to operate with the dataset.
3. Obtain a Wikipedia taxonomy over a domain.
4. Test some clustering methods.
5. Generate and test visualization tols.
6. Evaluate the resulting taxonomy.

Chapter 2

State of the art

The process of building a taxonomy by automatic means over an arbitrary domain is called **taxonomy induction**. This chapter summarizes the state of the art in the area as well as ontology evaluation using as dataset the Wikipedia encyclopedia.

2.1 Overview

Research in Artificial Intelligence (AI) has made tremendous progress in the last decades by employing data-driven techniques for solving tasks of ever increasing difficulty. However, working on knowledge-intensive applications such as semantic web technologies and question answering engines calls for complementing statistical methods with semantically rich representations based on world and encyclopedic knowledge, thus bringing the *knowledge acquisition bottleneck problem*, the difficulty to actually model the knowledge relevant for the domain in question, into focus yet again [33].

The problem when simulating human intelligence is the need for wide-coverage bases. As they are manually created, they are domain dependent or have a limited coverage [33]. Examples of widely used taxonomies are: WordNet (a large lexical database of English)[25, 34], Mesh (the U.S. National Library of Medicine's controlled vocabulary thesaurus)[45] or OpenCyc (the world's largest and most complete general knowledge base and commonsense reasoning engine)[9]. Other problem, related to the manpower factor is the difficulty of maintenance in rapidly changing domains which makes them hard to build with consistency [20].

This scenario motivates the following question.
How can one induce the taxonomic organization of concepts in a given domain starting from scratch?[20].

Terminology

This is the basic terminology that can be found in a concept network as described by [15].

term: An English word (for our current purposes, a noun or a proper name).

seed term: A word we use to initiate the algorithm.

concept: An item in the classification taxonomy we are building. A concept may correspond to several terms (singular form, plural form, the term's synonyms, etc.).

root concept: A concept at a fairly general (high) level in the taxonomy, to which many others are eventually learned to be subtypes/instances of.

basic-level concept: A concept at the "basic level", corresponding to the (proto)typical level of generality of its type.

instance: An item in the classification taxonomy that is more specific than a concept; only one example of the instance exists in "the real world" at any time. For example, Michelangelo is an instance, as well as Mazda Miata with license plate 3HCY687, while Mazda Miata is not.

classification link: A single relation, that, depending on its arguments, is either is a *type-of* (when both arguments are concepts), or is an *instance-of* or is an *example-of* (when the first argu-ment is an instance/example of the second).

2.2 The Wikipedia encyclopedia

Wikipedia is a free-access, free content Internet encyclopedia, supported and hosted by the non-profit Wikimedia foundation. Almost anyone who can access the site can edit almost any of its articles. Wikipedia is the sixth-most popular website and constitutes the Internet's largest and most popular general reference work.[47].

Wikipedia facts [53]:

- There are currently 33,780,664 articles in the Wikipedia; 4,599,383 of them in the english version.
- The figure 2.1 tries to illustrate how big the English-language Wikipedia might be if the articles (without images and other multimedia content) were to be printed and bound in book form. Each volume is assumed to be 25 cm tall, 5 cm thick, and containing 1,600,000 words or 8,000,000 characters. The size of this illustration is based upon the live article count.
- The figure 2.2 shows the number of articles of the english wikipedia (thick blue line) compared with a Gompertz model that leads eventually to a maximum of about 4.4 million articles (thin green line).
- The figure 2.3 compares the growth of the ten largest Wikipedias. The sum includes all 270+ Wikipedia languages.

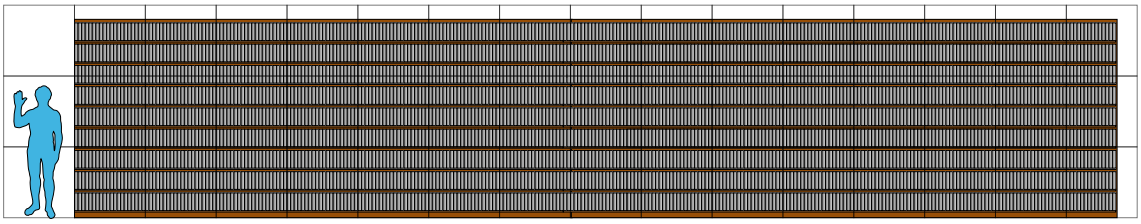

Figure 2.1: Estimated size (August 2010) of the printed english Wikipedia, 2036 volumes and 11 stacks.

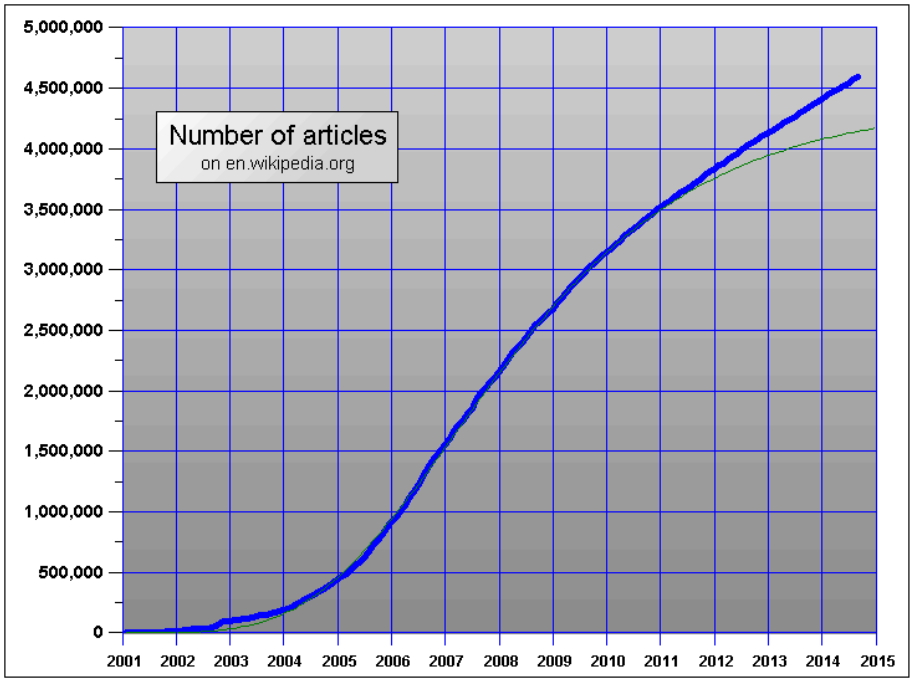

Figure 2.2: Actual articles (blue) versus Gompertz model (green).

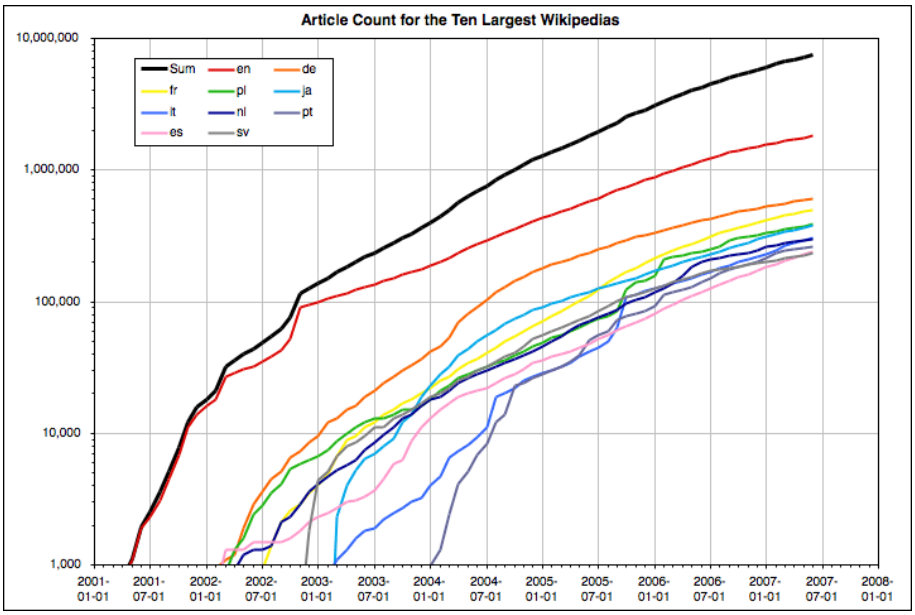

Figure 2.3: Top 10 Wikipedias comparison.

2.2.1 Dataset

Wikipedia has only existed since 2001 and has been considered a reliable source of information for an even shorter amount of time, researchers in NLP have just recently begun to work with its content or use it as a resource. Wikipedia has been successfully used for a multitude of AI and NLP applications. These include both preprocessing tasks such as named entity and word sense disambiguation, text categorization, computing semantic similarity of texts, coreference resolution and keyword extraction, as well as full-fledged, end-user applications such as question answering, topic-driven multi-document summarization, text generation and cross-lingual information retrieval.[33]

This are the main points for taking the Wikipedia as dataset when facing AI problems:

- It is a huge encyclopedia, and due its collaborative nature, the amount of pages increases everyday.
- It is up to date.
- While the contribution of any individual user might be imprecise or inaccurate, the continual intervention of expert contributors in all domains results in a resource of the highest quality [11].
- Provides an acceptable coverage on most popular domains.
- It is downloadable and free to use [51].
- Articles are written under the Neutral Point Of View criteria, which means representing fairly, proportionately, and, as far as possible, without bias, all of the significant views that have been published by reliable sources on a topic [52].

Because of this, Wikipedia is a good representation of the current human knowledge.

From a computational linguistics perspective, knowledge bases for NLP applications should be:

- Domain independent.
- Up-to-date.
- Multilingual.

Wikipedia fulfils all this requirements.

2.2.2 Categorization

This subsection has been greatly inspired by the article: *Tagging wikipedia: Collaboratively creating a category system* [44].

Although Wikipedia started as a set of articles, two years after the beginning, the community decided to create a category system to organize and tag the content of the site. This is done by assigning articles to categories through links in a similar way to the metadata practice of adding a tag to a piece of content.

In the field of information science, **knowledge organization (KO**) theorizes, analyzes and critiques systems designed to organize information. From a KO perspective, the Wikipedia category system can be seen as a thesaurus built through collaborative tagging. Hierarchies appear when adding categories to other categories, and as a result, an implicit concept network raises.

Relationships

This are the relationships that can be found in a taxonomy [42].

- **Associative**: a relationship between or among terms that leads from one term to other terms that are related to or associated with it. An associative relationship is a related term or cross-reference relationship.
- **Equivalence**: a relationship between or among terms in a controlled vocabulary that leads to one or more terms that are to be used instead of the term from which the reference is made. An equivalence relationship is a *used-for* term relationship.
- **Hierarchical**: a relationship between or among terms in a controlled vocabulary that depicts broader (generic) to narrower (specific) or whole-part relationships. A hierarchical relationship is a broader term to narrower term relationship. To understand the use of this hierarchical structure, it is mandatory to know the types of hierarchical relationships [2].
  - **Generic**: a generic hierarchical relationship is defined as a conceptual transitive closure. There are very few examples of this in the category system. But there are other indexing tools in Wikipedia that are organized in this way, for example, the Wikipedia page for "List of birds". That is, if we take the class to be "birds" all of the pages for which links are supplied in the list are pages for birds.
  - **Whole-part**: this relationship consists of a single concept or entity as the class with parts of that concept or entity as the subclass. An example of this type of hierarchical relationship in Wikipedia would be the pages listed under the category "States of the United States". Other than the page "U.S. state", the other pages under this category are all parts of the category itself.
  - **Instance**: these are general concepts or classes which have specific instantiations as a subclass. It is difficult to find examples of categories for which the subcategories are all instances of the category. This is more often achieved through lists in Wikipedia. An example of this type of hierarchical relationship in Wikipedia would be the "List of cathedrals" page. If we take "cathedrals" to be the class, all of the cathedrals listed on that page are instances of the class.
  - **Polyhierarchial**: this type of relationship describes cases when one term is located underneath more than one category. Many categories in Wikipedia are located in more than one parent category. Polyhierarchy is very common in the category system of Wikipedia.

  All four types of hierarchy relationships are in use in the category system of Wikipedia.

Purpose

The fact that the relationships between supercategories and subcategories in Wikipedia include both hierarchical relationships as well as associative relationships is due to the fact that the category system emerged from a community in which there were divergent views of what the system should look like.

- One of the editors who contributed to the discussions in the dataset stated the need for hierarchical and associative relationships.

```
So I think we need a way of distinguishing between a category where (a) you are
 asserting that everything in the category is an example of the thing it is in
     (ie list categories), and (b) categories where you are just providing
                     hierarchical links for convenience.
```

- A second editor said that a page might need multiple category designations. A clear articulation of the need for polyhierarchy.

> I'm thinking about some of the dog topics. For example, dog is a member of pets; dog is also a member of mammals; both mammals and pets are members of animals but neither is a subcategory of the other. Now, how about dog agility? It needs to go under the dog sports category, which needs to be under the dog category, because it's related to dogs. It also needs to go under the sports category, because it's a sport. It probably also needs to go under the hobby category. But dog and sports do not at any higher point in the hierarchy have a common parent.

- Other editors felt that the work of scoping the category system of Wikipedia was such a large task that it should be modeled on existing structures for information organization. One editor brought up the challenge of making relationship types explicit and suggested modeling the category system on the Resource Description Framework (RDF).

> The fix is to label the arrows: describe the relations. This is, in my limited understanding, what RDF does. That uses the terms subject, predicate, and object. The subject is the thing you're categorizing. The object is the category you're adding it to. And the predicate describes the relation. Predicates allow you to make semantic inferences programmatically.

Some members of the community felt that the amount of effort that was being expended in the design of the category system could be reduced if the purpose of the category system were explicitly articulated.

> A question concerning the purpose of categories: Is the primary purpose of categories to: Aid the reader in finding material that may be of interest, or relevant to a particular topic? Producing a taxonomy; wherein being included in one or more categories is an indication-nay, a declaration by the Wikipedia community-that the subject of an article is an instance of the category it is included in. I seem to suspect the latter...

> There seems to be a dichotomy between those who are looking to hone categories into encyclopedic taxonomies and those who are looking for a tagging system in which they can do keyword searches. The more we push at removing overcategorization, the more there is a need for a simpler tagging system. If we can answer that need, it might make everyone happier.

2.2.3 Problems

The lack of consensus with the purpose of the categorization system can be summarized in two points of view. On one hand, there is the **searching purpose**, that is, the user starts navigating at a top category and gets down in the hierarchy to the desired article. This is the most taxonomy-related approach of the structurey. On the other hand, there is the **browsing purpose**. The structure is used as a search-for-related-things tool. They navigate through the structure aiming to find related content, but the aspects of the relationships can vary. This view is related to the polyhierarchial relationships and will be the problem when converting a concept network into a taxonomy.

Again, the view of categories as a taxonomy could be against the NPOV policy and, therefore destined to fail [44]. This is a policy the editors take seriously.

Other underrated problem is the need for visualization tools. While Wikipedia has some extensions that allow for the exploration of the category system, there is nothing that specifically visualizes the assumed or real relationships among category nodes. A tool that in some way displayed the relation among categories would help regular users navigate with the category system and help individuals who wanted to tag pages [44].

2.3 Taxonomy induction

Recalling the problem of how to organize the gathered knowledge when there is no initial taxonomy, the automatic methods meant to build a taxonomy from scratch share two common stages:

1. **Term an relation extraction.** The produced output typically contains flat lists of terms and general relation types (term1 *is-a* term2). There are two approaches:
   - **Clustering** approaches are fully unsupervised and discover relations that are not directly expreseed in text. Their main drawback is that the may or may not produce the term types and granularities useful for the user.
   - **Pattern bassed** approaches harvest information with high accuracy, but they require a set of seeds and surface patterns to initiate the learning process.
2. **Taxonomy induction**. Consists on setting the relations between terms so the induced structure is a taxonomy.

Taxonomizing the terms is a very powerful method to leverage added information. Subordinated terms (hyponyms) inherit information from their superordinates (hypernyms), making it unnecessary to learn all relevant information over and over for every term in the language. But despite many attempts, no "correct" taxonomization has ever been constructed. Typically, people build term taxonomies (and/or richer structures like ontologies) for particular purposes, using specific taxonomization criteria. Different tasks and criteria produce different taxonomies, even when using the same basic level concepts. This is because most basic level concepts admit **multiple perspectives**, while each task focuses on one, or at most two, perspectives at a time [20]. This is a manifestation of the polyhierarchial relationship.

Attempts at producing a single multi-perspective taxonomy fail due to the complexity of interaction among perspectives, and people are notoriously bad at constructing taxonomies adherent to a single perspective when given terms from multiple perspectives. This issue and the major alternative principles for taxonomization are discussed in [14].

According to the the source of data, there are two alternatives to generate a taxonomy by automatic means.

2.3.1 Previous structure

The starting point is a concept network structure (i.e. the Wikipedia categorization system) and the aim is to expand it with new concepts or to adjust the relations to get a tree-like structure, a taxonomy. The benefit of this alternative is that there is no need to perform the term and relation step, so the effort will be on the taxonomy induction step by refining relations. This are the two main approaches.

Semantic based

This approach tries to mimic the human-decision-taking process by filtering concepts and relationships with regular expressions. This is the most common and best performing way to induce taxonomies, but since they are language dependant and, therefore, require a set of rules based on the vocabulary as well as the gramatic, they cannot be considerated a fully automated method but a semi-supervised one. Great work has been done in the English language. According to [32, 33] the main methods of this approach are (note the order importance of the methods):

1. **Category network cleanup**
   This preprocessing step cleans the network from meta-categories by removing all nodes whose labels contain any of the following strings: `wikipedia, wikiprojects, lists, mediawiki, template, user, portal, categories, articles, pages`.
2. **Refinement link identification**
   This preprocessing step identify *is-refined-by* relationships. It can be done in two ways:
   - Taking all categories containing "`by`" in the name and label all links with their subcategories.
   - Identify category pairs that match whith the patterns `X Y` and `X by Z` (i.e. Miles Davis Albums and Albums by Artist).
3. **Syntax-based methods**
   The first set of processing methods to label relations between categoris as *isa* is based on string matching of syntactic components of the category labels.
   (a) **Head matching**
   Given the lexical heads for a pair of categories, a category link *isa* is stablished if the two categories share the same head lemma, as determined by a finite-state morphological analyzer (i.e. `British Computer Scientists` *isa* `Computer Scientists`).
   (b) **Modifier matching**
   A link between categories is labeled as a *notisa* relationship if the stem of the lexical head of one of the categories occurs in a modifier position in the other catgory (i.e. `Crime Comics` *notisa* `Crime`).
4. **Connectivity-based methods**
   The connectivity-based methods utilizes the structure and connectivity of the categorization network to provide *isa* links in cases where relations are unlikely to be found in free text.
   (a) **Instance categorization**
   This method labels *instance-of* relationships heuristically by determining wether the head of the page category is plural.
5. **Lexico-syntactic basic methods**
   A majority voting strategy is used on pattern matching to find *isa* and *notisa* relationships. This patterns are highy language dependant, see [33].
6. **Inference-based methods**
   The last set of methods propagates the previously found relations by means of multiple inheritance and transitivity.
   (a) **Multiple inheritance propagation**
   First, propagate all *isa* relations to those super-categories whose head lemmas match the head lemma of a previously identified *isa* super-category.

(b) **Transitivity propagation**
Second, propagate all *isa* links to those super-categories which are connected through a path found along the previously discovered subsumption hierarchy, thus taking the transitive closure of the *isa* relation.

Graph based

This approach uses the structure of the concept network rather than the node names to set relations.

1. **Cycle removal**
   This method deletes al the cycles in the graph.

2. **Transitive Reduction** [1]
   The transitive reduction of a directed graph is a graph with as few edges as possible that has the same reachability relation as the given graph.
   These are some interesting properties:

   - The transitive reduction of a graph is unique.
   - The method removes redundant relationships that can be infered by the transitive property.

3. **Longest path** [20]
   This method surmount the problem in which multiple paths can be taken to reach from one concept to another. Intuitively, finding the longest path is equivalent to finding the taxonomic organization of all concepts.

Hybrid

This methods mix the two previous approaches.

1. **Redundant categorization** [33]
   This method tags the relations between two directly connected categories as *isa* if there is at least one page categorized in both categories and the category labels both have a plural head .

2.3.2 Corpus of documents

This alternative takes a corpus of documents under the taxonomy domain and uses their content to extract concepts, relationships and induce a taxonomy or enrich a previous one.
According to the document interpretation, two approaches can be taken.

Similarity based

This methods are characterized by the use of a similarity/distance measure to compute the pairwise similarity/distance between two documents, represented as word or term vectors, in order to decide if they can be clustered together or not.
The common steps they share are:

1. The construction of a *set of terms*, usually called Bag of Words or BOW. The words are extracted from the documents and mostly ever are: nouns, verbs, adverbs and adjetcives. Then they are stemmed so that the lexical root is taken as the term, this way the generalization power is guaranteed.

2. The *term weighting*, usuallly done with the Term Frequency-Inverse Document Frequency (TFIDF) [39] algorithm. At this point there is a document-term matrix, a vector space model [40] representing the documents as a set of weightened terms.

3. The *clustering of terms*. A process where is decided which terms belongs to the same group owing to the similarity between them.

An example of this technique can be found in [38], where the authors map Wikipedia articles to WordNet synsets. They compare the results obtained by stemming or not the terms and using the dot product versus the cosine as the similarity measure in the vector space model.
The BOW approach is not meant to induce a taxonomy from scratch, as a notorious disadvantage of this model is that it ignores the semantic relationship among words.

The size of the bag of words is frequently huge, up to thousands of terms, the easiest and mostly accepted way to deal with this manifestation of the *curse of dimensionality* is selecting only the top *k-terms*. Nevertheless there are other methods to reduce the dimensionality [22].

- **Feature extraction**
  It is a process that extracts a set of new features from the original features through some functional mapping such as principal component analysis (PCA) and word clustering. The downside is that the generated new features may not have a clear physical meaning so the clustering results are difficult to interpret.

- **Feature selection**
  It is a process that chooses a subset of terms from the original feature set according to some criterion. The selected feature retains the original physical meaning and provides a better understanding for the data and learning process. Depending on if the class label information is required, feature selection can be either unsupervised or supervised.
  This is a small summary covering some of this criteria [22]:

  - **Information Gain**
    A supervised method that measures the number of bits of information obtained for category prediction by the presence or absence of the term in a document.
  - χ^2 **statistic (CHI)**
    A supervised method that measures the association between the term and the category.
  - **Document Frequency**
    An unsupervised, simple but flexible, method that counts the number of documents in which a term occurs. It scales linerarly to large datasets.
  - **Term strength**
    It is computed based on the conditional probability that a term occurs in the second half of a pair of related documents given that it occurs in the first half. Since it is necesary to calculate the similarity for each document pair, the time complexity is quadratic to the number of documents. It is an unsupervised method.
  - **Entropy-based Ranking**
    The term is weightened by the entropy reduction when it is removed. The most serious problem of this method is its high computation complexity $O(MN^2)$. It is impractical when there is a large number of documents and terms, and therefore, sampling technique is used in real experiments.
  - **Term Contribution**[22]
    The document frequency method assumes that each term is of same importance in different

documents, it is easily biased those common terms which have high document frequency but uniform distribution over different classes. Term contribution is proposed to deal with this problem.

A remarkable work comparing different representations (TF-IDF, Latent Semantic Indexing and multi-word) for documents can be seen at [54].

Graph based

A corpus of documents can also be clustered by the analysis of the network that, through hyperlinks, connect them. This are the most relevant algorithms:

1. **Edge betweenness and clustering coefficient**
   The assumption is that edges lying in most of the shortest paths in the graph or with low clustering coefficient are likely to connect separate communities. By recursively deleting the edges with larger betweenness or low clustering, the graph splits into its communities.

2. **Network modularity optimization**
   They form cluster of nodes so that the density of link within the communities are maximized against the number of links among communities.

3. **Spectral methods**
   They are based on the analysis of the eigenvalues and eigenvectors of suitably chosen functions of adjacency matrix.

4. **MLC algorithm**[7]
   It detects strongly interconnected communities of nodes in a network by finding the attraction basins of random walks on the graph. It provides a fast response in a reasonable time even for networks including thousands of nodes, and can be tuned opportunely in order to maintain its efficiency.

Unfortunately, all the methods except the last one can only be applied in small graphs as they turn unusable in larger networks since they require exceeding computational resources of time. Even the MLC algorithm is unable to cluster large Wikigraphs.
The conclussion of [7] have great implications on considering links as a valid similarity measure.

> `The varying agreement between clustering and categorization across the studied versions of Wikipedia suggests that links in Wikipedia do not necessarily imply similarity or relatedness relations. From a technological point of view, this observation implies that, before switching to automatic categorization of items in Wikipedia and in other information networks, it should be tested how the selected clustering algorithm performs with respect to manual indexing.`

Set theoretical

This approaches partially order the objects acording to the incusion relations between their attribute sets. The most commonly used method is the Formal Concept Analysis or FCA [35], that relies on lattices to represent data. Works ave been done in taxonomy induction [30] and concept clusterization [8].

2.4 Ontology evaluation

Although the terms taxonomy and ontology are indistinctly used, that is a misnomer because ontologies are a generalization of taxonomies. Ontologies label relationships (*is-a* relationships are the only ones in taxonomies) and therefore elements can play different roles; there is no need for a hierarchy between elements so, if a taxonomy can be seen as a "tree", an ontology is often more of a "forest". An ontology might encompass a number of taxonomies, with each taxonomy organizing a subject in a particular way [27]. Ontologies are likely to classify words more carefully, perhaps as parts of speech, which human language, how precisely one word is the exact synonym for another, etc.

Here are two more formal definitions.

Taxonomy [31]

A taxonomy is a collection of controlled vocabulary terms organized into a hierarchical structure. Each term in a taxonomy is in one or more parent-child relationships to other terms in the taxonomy. There may be different types of parent-child relationships in a taxonomy (e.g., whole-part, genus-species, type-instance), but good practice limits all parent-child relationships to a single parent to be of the same type. Some taxonomies allow polyhierarchy, which means that a term can have multiple parents. This means that if a term appears in multiple places in a taxonomy, then it is the same term. Specifically, if a term has children in one place in a taxonomy, then it has the same children in every other place where it appears.
A taxonomy has additional meaning specified via whatever the meaning of the hierarchical link is. In a traditional "taxonomy" the meaning is generalization/specialization or "is a kind of", depending on what direction you are going. These days the word "taxonomy" is used to refer to other kinds of hierarchies with different meanings for the links (e.g., part of, broader topic than, instance of). Sloppy taxonomies will not identify explicitly what the meaning of the link is, and there may be different meanings. If a taxonomy has a variety of very carefully defined meanings for the hierarchical link, then it bears a stronger resemblance to an ontology.

Ontology [31]

A formal ontology is a controlled vocabulary expressed in an ontology representation language. This language has a grammar for using vocabulary terms to express something meaningful within a specified domain of interest. The grammar contains formal constraints (e.g., specifies what it means to be a well-formed statement, assertion, query, etc.) on how terms in the ontology's controlled vocabulary can be used together.
The word "ontology", when used in the AI/Knowledge Representation community, tends to refer to things that have a rich and formal logic-based language for specifying meaning of the terms. Both a thesaurus and a taxonomy can be seen as having a simple language that could be given a grammar, although this is not normally done. Usually they are not formal, in the sense that there is no formal semantics given for the language. However, one can create a model in UML and a model in some formal ontology language and they can have identical meaning. It is thus not useful to say one is an ontology and the other is not because one lacks a formal semantics. The truth is there is a fuzzy line connecting these things.

The clarification is important because in information science, the study field that deals with knowledge representation, the term "ontology" is used rather than "taxonomy", and for this reason most of related studies names articles after it.

The hard question is: *What is a good ontology?*
Good ontologies are the ones that serve their purpose. Complete ontologies are probably more than what most knowledge services require to function properly. The biggest impediment to ontology use

is the cost of building them, and deploying "scruffy" ontologies that are cheap to build and easy to maintain might be a more practical and economical option. Equally there has been much focus on the potential of ontology re-use, which would also lower the entry cost. In both cases, the existence of appropriate evaluation methodologies is essential.[5]

Ontology evaluation can be as complex as the taxonomy induction issue, in fact, these two fields are different points of view of the same problem, knowledge representation. Because of that, they share methods that apply slightly differently. There are two alternatives when evaluating an ontology, taking it as a whole and compare it against some kind of valid reference (which is the classical approach), or focusing in some parts and testing them independently.

Once distinctions have been done, it is time to review the methods used to measure the quality of induced taxonomies, which will the same ones used for ontologies.

2.4.1 Whole evaluation

This set of methods take the ontology as a whole and evaluate it against some kind of authority or purpose. This is the standard approach.

Human

In this qualitative approach, the evaluation is done by human means, generally by experts in the field who asses how well the ontology meets a set of predefined criteria, standards, requirements, etc. From a human point of view, the quality of the result is the best one of all the methods, at the expense of lacking automatization and consequently beeing unsuitable for arbitrary domains.
The biggest problem here is that it is quite hard to determine who the right users are, and what criteria to propose they use for their evaluation. Should the domain experts be considered the users, or the knowledge engineers, or even the end users? Should they evaluate an ontology more highly because it is "sensible", "coherent", "complete" or "correct", and what do we mean by these terms? Furthermore, most users could not evaluate the logical correctness of an ontology.

Construction criteria

A closely related qualitative approach would be to evaluate an ontology from the perspective of the principles used in its construction. While some of these design principles are valid in theory, it is extremely difficult to construct automated tests which will comparatively evaluate two or more ontologies as to their consistent use of "identity criteria" or their taxonomic rigour. This is because such principles depend on an external semantics to perform that evaluation, which currently only human beings are capable of providing. Furthermore, there is a significant danger that in applying a principles-based approach to ontology construction the result could be vacuous and of no practical use.

Golden standard

In this case, the authority is a prevously built knowledge base considered a good representation of the concepts in the domain. In the literature, almost always the golden standard is another ontology such as: WordNet[34], MeSH[45] or CyC/OpenCyc[9].

Corpus of documents, data driven

This set of methods involve comparisons of the ontology with a source of data (a collection of documents) to measure the congruence between the ontology and a domain of knowledge.
The problem here is that if the results differ from the gold standard, it is hard to determine whether that is because the corpus is inappropriate, the methodology is flawed or there is a real difference in the knowledge present in the corpus and the gold standard. In any case, this approach is more applicable when one is trying to evaluate ontology learning methodologies. In the Semantic Web scenario, it is likely that one has to choose from a range of existing ontologies the most appropriate for a particular domain, or the most appropriate to adapt to the specific needs of the domain/application.
This is an architecture for ontology-corpus evaluation proposed by [5]:

1. **Indentifying keywords/terms**
   This is essentially a form of automated term recognition, and thus the whole panoply of techniques existing can be applied: TF-IDF, LSA, PLSA [13], etc; and a clustering method.

2. **Query expansion**
   Because a concept in the ontology is a compact representation of a number of different lexical realisations in a number of ways, it is important to perform some form of query expansion of the concept terms. It can be done using WordNet to add two levels of hypernyms to each term in a cluster. There are other ways to expand the a term using (foexample) IR techniques.

3. **Ontology mapping**
   Finally, the set of terms identified in the corpus need to be mapped to the ontology.

Given a corpus appropriately annotated against an ontology, we could count how many concept terms in the ontology match those lexical items that have been marked up. This would yield initial (crude) measures of lexical keyword coverage by ontology labels (precision and recall). This provides figures which reflect the coverage of the ontology of the corpus. The most common scenario is one where there are items absent as well as items unneeded.

The advantage of using a cluster analysis approach is that it permits the creation of a measure of structural fit. We can imagine two ontologies with identical concept sets which, however, have the concepts differently organised and thus concepts are at a different distance from each other. Thus the authors propose a 'tennis measure' [41] for an ontology which evaluates the extent to which items in the same cluster are closer together in the ontology than those in different clusters. What is determined as close is dependent on the probability model used to derive the clusters.

The authors express the evaluation of the "best fit" between a corpus and one among a set of ontologies as the requirement of finding the conditional probability of the ontologies given the corpus. The ontology that maximizes the conditional probabilityof the ontology O given a corpus C is then the best fit ontology O^*:

$$O^* = argmax_O P(O|C) = argmax_O \frac{P(C|O)P(O)}{P(C)}$$

Performance

The measure is given the results of using the ontology in an application. From an utility perspective this is the best approach, but the ontology could not have meaning for humans, it could not be a reasonable knowledge representation.

2.4.2 Level-based evaluation

An ontology is a fairly complex structure and it is often more practical to focus on the evaluation of different levels of the ontology separately rather than trying to directly evaluate the ontology as a whole. This is particularly true if we want a predominantly automated evaluation rather than entirely carried out by human users/experts. Another reason for the level-based approach is that when automatic learning techniques have been used in the construction of the ontology, the techniques involved are substantially different for the different levels. The individual levels have been defined variously by different authors, but these various definitions tend to be broadly similar and usually involve the following levels [4]:

Lexical, vocabulary, concept and data level

The lexical content of an ontology can be evaluated using an Information Retrieval approach. This approach (applied in [20]) use concepts such as precision and recall against a gold standard.
This are the methods:

- *Precision* is the percentage of the ontology lexical entries or concepts that also appear in the golden standard, relative to the total number of ontology words.

$$Precision = \frac{|O \cap G|}{|O|}$$

- *Lexical Recall* (LR) is the percentage of golden standard lexical entries that also appear as concept identifiers in the ontology, relative to the total number of golden standard lexical entries.

$$Recall = \frac{|O \cap G|}{|G|}$$

Where:

O**:** concepts found in the ontology to evaluate.

G**:** concepts found in the golden standard.

A way to achieve a more tolerant matching criteria is to augment each lexical entry with its hypernyms from WordNet or some similar resource. Then, instead of testing for equality of two lexical entries, one can tet for overlap between their corresponding set of words (each set containing an entry with hypernyms)[4].

Hierarchy, taxonomy and other semantic relations

This methods compare an automatically generated mapping (i.e. an induced taxonomy) against a gold standard. They focus on the taxonomy edges (i.e. in a Wikipedia taxonomy, *isa* relations between categories).
This will be the nomenclature:

E_O : edges of the ontology to evaluate.

E_G**:** edges of the golden standard.

Metrics:

- *Coverage* measures the size of the intersection between the candidate and the golden standard. It is a precision measure using relations instead of concepts.

$$Coverage = \frac{|E_O \cap E_G|}{|E_G|}$$

- *Novelty* is the rate of *isa* pairs in the candidate that have no mapping in the golden standard.

$$Novelty = \frac{|E_O - E_G|}{|E_O|}$$

- *Extra Coverage* measures the "gain" in knowledge provided by the candidate with respect to the existing bases by calculating the proportion of unmapped category pairs *isa* relation to the total number of semantic relations in the gold standard.

$$ExtraCoverage = \frac{|E_O - E_G|}{|E_G|}$$

Other similarity measures as seen in [8, 23].

- *Semantic Cotopy* (SC)
- *Taxonomy Overlap* ($\bar{TO}$)
- *Relation Overlap* ($\bar{RO}$)
- *F-Measure*

$$F = \frac{2 * LR * \bar{TO}}{LR + \bar{TO}}$$

In [23], the authors propose several measures for comparing the relational aspects of two ontologies. Although the need for a golden standard when evaluating an ontology is in a way a drawback, an important positive aspect is that once defined, comparison of two ontologies can proceed entirely automatically.

Context level

Sometimes the ontology is a part of a larger collection of ontologies that may reference one another. This context can be used for evaluation of an ontology in various ways.
It is possible to use cross-references between semantic-web documents to define a graph and then compute a score for each ontology using for example the PageRank algorithm [29]. As relationships between ontologies can be labeled different, there will be several networks and therefore several ontology rankings to choose depending on the needs.

Application level

This is the same as the performance approach in the whole level. A good ontology is one which helps the application in question produce good results on the given task. This is elegant in the sense that the output of the application might be something for which a relatively straightforward and non-problematic evaluation approach already exists.
This evaluation method has several drawbacks:

- An ontology is good or bad when used in a particular way for a particular task, not by itself alone.

- Then effect of the ontology in the application could be small or indirect, so the outcome is not meaningful.
- Comparing different ontologies is only possible if they can all be plugged into the same application.

Syntactic level

This is mostly used for manually constructed ontologies. The ontology is usually described in a particular formal language and must match the syntactic requirements of that language. Various other syntactic considerations, such as the presence of natural-language documentation, avoiding loops between considerations, etc., may also be considered.

Structure, architecture, design

This is primarily of interest in manually constructed ontologies. The ontology must meet certain predefined design principles or criteria; structural concerns involve the organization of the ontology and its suitability for further development.

Multi-criteria approaches

Another family of approaches to ontology evaluation deals with the problem of selecting a good ontology (or a small short-list of promising ontologies) from a given set of ontologies, and treats this problem as essentially a decision-making problem. To help us evaluate the ontologies, we can use approaches based on defining several decision criteria or attributes; for each criterion, the ontology is evaluated and given a numerical score. An overall score for the ontology is then computed as a weighted sum of its per-criterion scores. A drawback is that a lot of manual involvement by human experts may be needed. In effect, the general problem of ontology evaluation has been deferred or relegated to the question of how to evaluate the ontology with respect to the individual evaluation criteria. On the positive side, these approaches allow us to combine criteria from most of the levels.[4]

2.4.3 Conclusions

Ontology evaluation remains an important open problem in the area of ontology-supported computing and the semantic web. There is no single best or preferred approach to ontology evaluation; instead, the choice of a suitable approach must depend on the purpose of evaluation, the application in which the ontology is to be used, and on what aspect of the ontology we are trying to evaluate. In our opinion, future work in this area should focus particularly on automated ontology evaluation, which is a necessary precondition for the healthy development of automated ontology processing techniques for a number of problems, such as ontology learning, population, mediation, matching, and so on.[4]

2.5 Related work

In this section, two articles that are highly related with thesis are exoposed. Although the original application of them does not seem to fit into the goals of this work, the methods they present can surely be tuned to enhace the previously exposed taxonomy induction and evaluation approaches.

2.5.1 Exploiting Wikipedia as External Knowledge for Document Clustering

This article [16] faces the problem of using a knowledge structure (i.e. an ontology) to improve the results of a text clustering algorithm.
A concern when clustering text (from a corpus of documents) is that using a bag of words approach leads to situations where two documents that represent the same topic are assigned to different groups just because they use different collections of words. A disadvantage of the BOW model is that it ignores the semantic relationships among words. To surpass this obstacle, the document representation (vector of terms) can be enriched with the aid of an ontology. The method involves a matching between the documents terms and the ontology concepts and performing one of this two actions:

1. Replace the terms with their matched topics. This can cause information loss if the ontology coverage is small.

2. Expand the document representation with the matched topics. This can introduce noise due to the dimensional increment and the need for sense disambiguation.

So, the identified needs are:

- An ontology which can cover the topical domain of individual document collections as completely as possible.

- A proper matching method which can enrich the document representation by fully leveraging ontology terms and relations but without introducing more noise.

For the first need, the authors take the Wikipedia encyclopedia, and for the second need, they propose a new document representation method.

Document representation

The purpose is to obtain a mapping between the corpus documents and their related Wikipedia categories. Some kind of multi-topic extraction.
The document-category mapping is done in three steps:

1. **Concept-category matrix**
   This matrix is created intuitively based on the connection between concepts and categories which is explicit in Wikipedia.

2. **Document-concept matrix**
   This step creates a vector of Wikipedia concepts per document. This are the propossed matching schemas.

   (a) *Exact-match scheme.* Each document is scanned to find Wikipedia concepts. To address the problem of synonymy, the redirect links found in Wikipedia are used so all the terms representing the same topic are redirected to the same article (a smart way to reduce dimensinality). The advantage is, exact-matching is very efficient but it produces good result only when Wikipedia has a good coverage of the phrases appearing in a dataset. Always has low recall.

   (b) *Relatedness-Match scheme* This technique is intented to improve recall of the exact-match scheme. It involves two steps:

i. The creation of a term-concept matrix from the Wikipedia article collection. Using TFIDF weights, the relatedness between the term (article) an each Wikipedia concept (words). This representation is cut at the top-5 concepts (per document).

ii. The word-concept matrix is used as a bridge to associate documents with Wikipedia concepts.

This method is more time consuming than the previous one. It help identify relevant Wikipedia concepts which are not explicitly present in a document and it is especially useful when Wikipedia concepts have less coverage for a dataset.

3. **Document-category matrix**
This matrix is derived from the concept-category matrix and the document-concept matrix with a join operation on the concept dimension. Then TFIDF is applied so it is possible to measure the similarity between any two documents represented as category vectors.

This new document features space model is merged with the existing document vector space model with the hope to improve the clustering results.

Clustering

Once the new features are computed, the authors test the hypotesis performing two types of clustering [18]:

- **Agglomerative clustering** consider each document as a cluster and repeatedly merge pairs of clusters with shortest distance until only one cluster is formed covering all the documents. Using standard vector cosine similarity as document similarity measure, both single linkage and average linkage suffer a severe chaining problem, so complete linkage is used as distance measure.

- **Partitional clustering** iteratively calculate the cluster centroids an reassign each document to th eclosest cluster until no document can be reassigned. Spherical K-means is the chosen algorithm. The distance from a document to a cluster centroid iis calculated based on the content similarity as well as concept similarity or category similarity, or both of them. Since the result depends on the initialization, the authors performs ten rounds with random initizalizations for each evaluation.

Experiments

The experiments are performed taking as features the seven different combinations of the extracted features: word, concept and category. The baseline is defined by word-only vectors.
In the agglomerative clustering, the best results are obtained from using a (word,category) or a (word,concept,category) scheme. But (word,concept,category) does not perform better than (word,category). Sometimes (word,concept) performs worse than the baseline. This indicates that integrating Wikipedia concept information into clustering process does not necessarily improve clustering performance. The results show that category information is more useful than concept information for improving clustering results. The authors think this could be as the Wikipedia collecion contains too much noise and they do not disambiguate concept senss during the concept mapping process.
In the partitional clustering, also the (word,category) and (word,concept,category) schemes obtained the best results, but the effect of category information and cluster information in the results is not as significant as in agglomerative clustering.

Conclusion

This article is intersting because some kind of hidden topic modeling is performed without naming it explicitly. Instead of taking a probabilistic approach, the topic modeling is performed with the aid of a taxonomy. This method can be applied to taxonomy induction to resolve polyhierarchical relationships and therefore, refine a concept network, obtained from the categorization system, with a copus of documents under the domain, Wikipedia articles, to obtain a tree-like taxonomy.

2.5.2 Semantic Simiarity Based on Corpus Statistics and Lexical Taxonomy

This article [19] proposes a method that combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributional analysis of corpus data. Specifically, the proposed measure is a combined approach that inherits the edge-based approach of the edge counting scheme, which is then enhanced by the node-based approach of the information content calculation.
The authors calculate the similarity between nodes in a taxonomy with a hybrid approach formed by this methods.

Node-based (Information Content) Approach

A node in the taxonomy represents a unique concept consisting of a certain amount of information, words w. This is clearly a data-driven approach. The similarity between two concepts is the extent to which they share information in common.
Based on the infomation theory, the information content (IC) of a concept/class c can be quantified as follows:

$$IC(c) = log^{-1}P(c)$$

Where $P(c)$ is the probability of encountering an instance of concept c. In the case of the hierarchical structure, where a concept in the hierarchy subsumes those lower in the hierarchy, this implies that $P(c)$ is monotonic as one moves up the hierarchy. As the node probability increases, its information content or its informativeness decreases. If there is a unique top node in the hierarchy, then its probability is 1, hence its information content is 0.
The similarity between two concepts can be formally defined as:

$$sim(c_1, c_2) = max_{c \in Sup((c_1,c_2))}(IC(c)) = max_{c \in Sup((c_1,c_2))}(-\log p(c))$$

Where $Sup((c_1, c_2))$ is the set of concepts that subsume both c_1 and c_2.

The concept probability can be computed using maximum likelihood estimation (MLE):

$$P(c) = \frac{freq(c)}{|c|}$$

And the frequency $freq(q)$ of a class can be computed in two ways:

- Simple approach.

$$freq(c) = \sum_{w \in words(c)} freq(w)$$

- Considering the number of word senses factor.

$$freq(c) = \sum_{w \in words(c)} \frac{freq(w)}{|classes(w)|}$$

Edge-based (Distance) Approach

The edge based approach is a more natural and direct way of evaluating semantic similarity in a taxonomy. It estimates the distance (i.e. edge length) between nodes which correspond to the concepts/classes being compared. Given the multidimensional concept space, the conceptual distance can conveniently be measured by the geometric distance between the nodes representing the concepts. Obviously, the shorter the path from one node to the other, the more similar they are.

The distance should satisfy the properties of a metric: zero property, positive property and triangular inequality. The simplest form of determining the distance between two nodes is the shortest path between them. The problem is, the distances between any two adjacent nodes are not necessarily equal. That is why edges should be weightened.
This are features that can be calculated to weight the edge according to the structural characteristics of the taxonomy.

- **Network density** describes the portion of the potential connections in a network that are actual connections [37]. The greater the density, the closer the distance between the nodes.
- **Node depth** means that the distance shrinks as one descends the hierarchy, since differentiation is based on finer and finer details.
- **Type of link** is the relation type between nodes. Hyponym/hypernym define *isa* relationships, meronym/holonym define *part-of, substance-of* relationships ...
- **Link strength** measures the closeness between a specific child node and its parent node, against those of its siblings. This is the place where corpus statistics could contribute (as stated by the authors).

In determining the overall edge based similarity, most methods just simply sum up all the edge weights along the shortest path.

Results

The authors combine the two approaches paying attention to the determination of the *link strength*. In the combined approach, they argue that the strength of a child link is proportional to the conditional probability of encountering an instance of the child concept c_i given an instance of its parent concept p.

$$P(c_i|p) = \frac{P(c_i \cap p)}{P(p)} = \frac{P(c_i)}{P(p)}$$

They apply their method against a human judgement and this are the results.

Similarity Method	Correlation(r)
Human Judgement	0.8848
Node Based	0.7941
Edge Based	0.6004
Combined approach	0.8282

It is impressive the combined approach nearly reaches the human judgement. From a machine learning perspective, this will be an ensemble model that follos the rule *"Learn many models, not just one"* [10].

This method should also a step in taxonomy induction. It worth the try.

Chapter 3

Proposal

3.1 A new framework

The original purpose of this thesis was to explore how the taxonomy induction process could be improved with text clustering. The previous chapter provides an undoubted expertise in the field and a good starting point. But, in order to propose experiments and methods, there are new issues that must be addressed before achieving the original goal.

Almost all the articles read have an experimental phase that guarantees the correctness of the tested hypothesis. However, there is a common problem when delving into the methodology; the impossibility of the replicating the experiments. That is, one wants to test their hypothesis and get the same results to compare methods, metrics or performances but it is almost impossible to do since some steps are vaguely described and therefore there is no way to know how do they obtain the results.

From the perspective of the productivity and Amdahl's law [21], it will be very interesting to have a way to replicate experiments accurately and test new hypothesis. Seeing that the investigator's toolbox is neven shared between the community (everybody knows the intentions behind this behaviour), the proposal of this thesis is a framework able to mitigate this situation.

3.1.1 Modern tools

With the evolution of the technology there is no point in keeping neither old tools nor methodologies if they can be replaced by others that increase the productivity. The main features the framework should provide are this ones:

- *Fast prototyping and good scalability*: with scripting languages such as Python [36].
- *Colaborative environment*: with CVS systems such as GitHub [12].
- *Interactive visualization*: based on standard platforms such as web browsers, and libraries like D3.js [3].
- *Interactive documentation*: so experiments and documentation can be seen together. A good example of this paradigm is the IPython Notebooks [17].

3.1.2 Layered architecture

The framework architecture is divided into six abstraction layers. Each one has a very specific purpose and improvements can be done with no coupling problems.

File layer : contains the dataset and index files.

Content layer : provides a page and category retrieval service.

Taxonomy layer : contains the taxonomy induction functionalities.

Similarity layer : contains the document similarity functionalities.

Visualization layer : provides an easy way to see the results.

3.2 File layer

This layer contains the files needed by the framework. This files are the Wikipedia dataset and an index structure over it to guarantee a fast information retrieval.

3.2.1 The Wikipedia dataset

In the previous chapter, the goodness of the Wikipedia encyclopedia as a good dataset was justified.

Working offline

There are two options when accessing the Wikipedia contents. The first one is on-line through the API [49]. Even though this is the easiest method, it is unsuitable for heavy-duty tasks unless we want to collapse the non-profit Wikimedia Foundation servers. On the other hand, there is the off-line option. It is possible to download all the contents in a few files and manage them on our own. This methodology have several benefits:

- Relieve the Wikipedia servers from intensive querying.
- Speed up your queries .
- Have an inmutable dataset to compare results.
- Independence from your network connection.

These advantages are provided at the following costs:

- Downloading the Wikipedia dump can be slow due to the size of the files.
- Building the indices takes time, it is a slow process.
- It is necesary to store all the data in your machine.

Fortunately, the first two actions are only executed once (at the start of the project) while for the third drawback, nowadays data storage has become extremely cheap.

Files

This are the files to download:

- **Pages**
  XML file that contains all the pages.
  - URL: http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2
  - Structure: http://www.mediawiki.org/xml/export-0.6.xsd
- **Categories**
  SQL file that contains the category names and identifiers.
  - URL: http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-category.sql.gz
  - Structure: http://www.mediawiki.org/wiki/Manual:Category_table
- **Categorylinks**
  SQL file that contains the association between categories and pages or other categories.
  - URL: http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-categorylinks.sql.gz
  - Structure: http://www.mediawiki.org/wiki/Manual:Categorylinks_table

3.2.2 Building indices

Once the files have been downloaded, it is time to build the indices over them. The program parses the original files to a CSV representation and, if needed (because the representation doesn't fit in memory as a python structure), transforms the CSV to a sqlite database. The process takes an hour or so in a modern computer. Since it is a sequential algorithm, speed depends on the hard drive I/O and the CPU capabilities (multicore is not used). Speed-ups can be achieved improving this components (i.e. using a SSD).

Pages

The file `enwiki-latest-pages-articles.xml` contains all the Wikipedia pages in a huge (45GB) XML file. The data is parsed into a CSV file and then converted to a Sqlite3 file.

The Sqlite3 file contains a table with one row per page with the following columns:

- **id**: numeric identifier of the page.
- **title**: title of the page
- **offset**: number of bytes from the beginning of the enwiki-latest-pages-articles.xml where the actual page can be found.

There are also built the following indices to speed up queries:

- **i_id**: index over the id field.
- **i_title**: index over the title field.

These are processing times:

Source	Output	Time (min)
xml_pages	csv_pages	25
csv_pages	sqlite_pages	5

Categories

The file `enwiki-latest-category.sql` contains information about categories, but only the id and the title are extracted into a CSV file. As long as the CSV file can be loaded into a native Python structure, there is no need to have this information stored in an external database.

This is the processing time:

Source	Output	Time (min)
sql_category	csv_category	1

Categorylinks

The file `enwiki-latest-categorylinks.sql` contains the parent relationships between categories, and categories with pages. It is parsed to a CSV file and then to a Sqlite3 file with the following columns:

- **child_id**: id of the children page.
- **parent_id**: id of the parent page.
- **type**: number that indicates the type of the children page.
    1. page.
    2. category.
    3. file.

There are also built the following indices to speed up queries:

1. **i_child**: index over the child_id field.
2. **i_parent**: index over the parent_id field.

These are processing times:

Source	Output	Time (min)
sql_categorylinks	csv_categorylinks	15
csv_categorylinks	sqlite_categorylinks	17

3.2.3 Parsing

For efficiency reasons, the original files are parsed with expat [43], a Simple API for XML or SAX parser [48]. Contrary to Domain Object Model or DOM parsers who treat the file as a whole, SAX ones go across the file sequentially and only keeps in memory what is needed for the task. In this case the files can be seen as bunch of articles or relationships, not as trees or other more machine demanding structures. That is why expat has turned out to be the perfect parser for the indexing process.

3.3 Content layer

This module provides an abstraction layer to interact with with the Wikipedia dump through the built indices in an offline way. It consists in two components, one for dealing with the pages retrieval and the other to extract relationships between pages.

3.3.1 Pages

This are the functionalities of the class:

- Retrieve the content of a page, identified by its title or numeric page identifier, as a dictionary. Because the content of a page is in a raw format called Markup [24], a postprocessing step is needed to extract links, raw text and other data. This behaviour can be controlled with a function.
- Get the title of a page from its nueric identifier or vice versa.

3.3.2 Categories

The main functionality of this class is to obtain the children pages of a category page. The result is a 2-tuple with a list of pageid,title list of pages and a pageid,title list of subcategories.

3.4 Taxonomy layer

This layer contains the taxonomy induction functionality. The starting point is the concept graph obtained from the Wikipedia categorization system and the goal is refine it to get a tree-like structure which will be the taxonomy. As the graph library, NetwokX [26] has been chosen.

3.4.1 Exploratory process

The main functionality is obtaining a concept graph. This is done by a Depth-First Search starting at an arbitrary root category and setting a maximum level of depth. This is a naïve graph, built from exploring the category relations without filtering anything. It gives an view of the subset extracted from the dataset.

3.4.2 Conflicts

The concept graph, also known as category graph, is unlikely to be a tree structure or a taxonomy. However, it is possible to automate some refinement steps in the induction problem.

Empty categories

This step removes the category nodes that have no articles as children. This is especially important when measuring a category by the information their children have.

Cycles

Even though it is said that the Wikipedia categorization system is a Directed Acyclic Graph [7, 50], experiments over the dataset have proved it is not true (this could be due to some kind of inconsistency). The solution to this conflict is pretty easy, as the concept graph has the edges labeled with the depth, removing the edge with the highest depth in the cycle solves the problem.

Multiple parents

This conflict appears when a node have several parents, this is a polyhierarchial relationship that is caused by the muti-perspective category tagging system. There is no trivial solution, but a set of automatic countermeasures are proposed to be used before reaching a manual solving.

Semantic cleanup
It consists on deleting category nodes according to some patterns as seen in 1. Although there are generic patterns, the problem is that this option is highly domain dependant as will be shown in experiments.

Transitive reduction
A path is removed if their nodes are a subset of the nodes of other path with the same origin and destiny as seen in 2. The formal method will be performing a transitive reduction over the graph [1].

Specification
It consist on setting as the correct relation the deepest one, this makes sense when the conflict is a category-page type.

Generalization
It consist on setting as the correct relation the highest one, this makes sense when the conflict is a category-category one.

3.4.3 Considerations

Other non trivial problem is the order the refinement steps should be applied. Removing empty categories, then cycles and at the end solve multiparent nodes seems to be the logical way, but while into the multiparent conflicts, the steps to take and in which order could affect excesively the result space. This high variance on the possible results will indicate a poor generalization power of the method set [10].

3.4.4 Other functionalities

This are other useful functionalities.

Conflict graph: builds a graph from the nodes that generate the conflict. It is useful to reduce the graph to the problem itself and visualize it later.

Node searching: retrieves the node identifier by performing a search over the node title. This are the matching rules, so once one is met the algorithm stops.

1. Full title matching.
2. Title starting matching.
3. Title tokenization and pseudo AND matching.
4. Title tokenization and OR matching.

Save/load: allow to save a graph and reload it later so it is not necesary to perform again an exploratory process.

3.4.5 Similarity layer

The purpose of this layer is to compute the similarity between Wikipedia articles. It is the data-driven contribution to the taxonomy induction. Computed similarities can have two applications:

- *Helping in the multiparent conflict.* When a polyhierarchial relationship is found, the tie can be solved by deleting the less related-to-the-category edge. The features to use for a category in order to calculate a similarity should be an average of their children article ones. Further investigation should be done in this area (article-to-category feature propagation).

- *Building alternative knowledge structures.* It is possible to perform a hierarchial clustering with the extracted features from the articles so a new structure, a dendrogram, is infered. Then, it is possible to compare the induced taxonomy against the dendrogram to see how they match at a structural level. As seen in 2.3.2, this is not and easy task and results should be interpreted cautiously as there are a lot of possible parametrizations.

The implementation of this layer is done in the class `Similarity`. It is initialized whith all the articles/documents under the domain of the taxonomy and holds a document-feature matrix to store the per-document computed features.

They key to success when performing machine learning, in this case clustering text, is having meaningful features that represent true relationships between the documents. Feature engineering is the key [10]. The framework proposes the following similarity methods:

1. TF-IDF. The basic steps of this method (mostly used as the baseline) are:
   (a) Convert the document to a suitable character set.
   (b) Split the document in words.
   (c) Remove stopwords. Words that do not add relevant semantic meaning (i.e. they appear in all documents).
   (d) Stem the words into terms, convert words to their lexical root.
   (e) Compute the document-term TF-IDF weight and keep the top-k terms.
   (f) At this point it is possible to measure the similarity between two documents represented as term vectors using the cosine similarity.
2. Shared links. The fact that two articles can link to the same page can be used as a similarity measure since that means their topics are related.
3. Wiki links. As one article can link to another, this feature can be used as a similarity measure.

3.5 Visualization layer

This layer provides the necesary means to visualize the data.
The approach taken is to use as the visualization tool an HTML5-compatible web browser (Mozilla Firefox or Google Chrome). The apearance of data-driven JavaScript libraries such as D3.js [3] means that there is no longer need to generate static figures as interactive plots enhace the exploratory process and allows the user to focus in the most interesting parts of the problem. Another feature is the ability to export the current view to a high-quality SVG image. These are the implemented visualization tools.

3.5.1 Wiki Graph

This web page takes a graph exported in the taxonomy layer as a JSON file, and shows it. The graph layout is force-directed and the parameters have been set to ease the visualization. This is the legend:

Blue circle : an article.

Blue square : a category.

Red node : a conflicting node, article (circle) or category (square).

This are the features:

- The mouse wheel controlls zooming.
- Clicking and dragging the background (any white area), controls the whole graph positioning.
- Clicking and dragging a node, the first time fixes it, the second time releases it.
- Clicking on SVG, opens a new tab so the graph can be saved as an SVG image.

3.5.2 Word Cloud

This web page takes the pairs (term, average TF-ID weight) JSON export from the tfidf similarity method an generates a word cloud plot. It is useful to see the relevant terms of a domain and it is also possible to configure some display options. It can also be exported as a SVG image.

Chapter 4

Experiments and results

This chapter shows some experiments and results found.

4.1 Cycle removal

Note: This section is the printed version of the interactive experiment presented as an IPython Notebook at `pykitools/doc/notebooks/Experiment - Cycle Removal.ipynb`.

It is said the Wikipedia categorization system is a directed acyclic graph [7, 50], yet other articles recognise this potential issue [20]. This experiment shows the existence of cycles between categories. This are the followed steps:

1. Generate a category graph with `Category:Political spectrum` as the root node and a maximum depth of 5 levels.
2. Analyze the graph and search for cycles.
   There is a cycle with two nodes between the categories:
   - `Socialism`. Appearing at levels 3 and 5.
   - `Economic planning`. Appearing at level 4.
3. The cycle has the form `Socialism`→ `Economic planning` ↪ `Socialism`→ `Economic planning` ...

As seen in 3.4.2 the solution consists on deleting the highest depth relation in the cycle which is the one between the the node with the highest depth and the lowest.

If the "Economic planning" Wikipedia category page http://en.wikipedia.org/wiki/Category:Economic_planning is observed, it is possible to see that there is no "Socialism" subcategory. So, the cycle found must be due to an inconsistency on the Wikipedia dump taken as the dataset. Anyway, this is a problem to be aware of.

4.2 Transitive reduction

Note: This section is the printed version of the interactive experiment presented as an IPython Notebook at `pykitools/doc/notebooks/Experiment - Transitive Reduction.ipynb`.

In the paper *A Semi-Supervised Method to Learn and construct Taxonomies using the Web* [20], in the subsection *3.3 Graph-Based Taxonomy Induction*, the authors identify 2 problems after creating the concept network:
After the concept positioning procedure has explored all concept pairs, we encounter two phenomena:

1. *Direct links between some concepts are missing.*

2. *Multiple paths can be taken to reach from one concept to another.*

The proposed solution is:
To surmount these problems, we employ a graph based algorithm that finds the longest path in the graph.
...
Intuitively, finding the longest paths is equivalent to finding the taxonomic organization of all concepts. First, if present, we eliminate all cycles from the graph. Then, we find all nodes that have no predecessor and those that have no successor. Intuitively, a node with no predecessors p is likely to be positioned on the top of the taxonomy, while a node with no successor s is likely to be located at the bottom.
...
For each (p, s) pair, we find the list of all paths connecting p with s. In the end, from all discovered candidate paths, the algorithm returns the longest one. The same graph-based taxonomization procedure is repeated for the rest of the basic level concepts and their hypernyms.

From their results:
To evaluate the performance of a taxonomy induction algorithm, one can compare against a simple taxonomy composed of 2–3 levels. However, one cannot guarantee that the algorithm can learn larger hierarchies completely or correctly.

The method proposed by the authors is to keep the longest path between two nodes as a multiparent conflict disambiguation. It is more destructive than a merely transitive reduction since it removes relationships that are kept by the later method.

This experiment shows that although a transitive reduction keeps the information relationship, the result could not be what is expected. This are the followed steps:

1. Generate a category graph with `Category:Political spectrum` as the root node and a maximum depth of 5 levels.

2. Remove cycles.

3. Set the origin node: `Category:Fascism`

4. Set the destiny node `Category:Nazism`

5. Find the possible paths:

   - `Fascism` → `Anti-fascism` → `Communism` → `Anti-communism` → `Nazism`
   - `Fascism` → `Nazism`

   Here we see that `Nazism` is a subcategory of `Fascism` and that are connected by two paths. The nodes of path are a subset of the other path, so a transitive reduction will keep all the information.

6. A transitive reduction is performed and the shortest path is removed since it is contained in the longest one.

The problem is that even though the longest path keeps all the relations, `Nazism` is not a subcategory of `Anti-fascism` or `Communism`, therefore this taxonomization is incorrect.

4.2.1 Semantic reduction

This particular case should not void the transitive reduction as a conflict solving method, but to warn the user that theoretical guarantees do not always fit perfectly with our purposes [10]. As an improvement for the transitive reduction, I propose a semantic reduction previous step.
The semantic reduction behaves similarly to the category network cleanup seen in 1, but instead of deleting categories, it takes a list of patterns and in the case of a multiparent conflict, the relations belonging to the path that contains any node that matches with the patterns are removed. To solve the case exposed in this experiment, some kind of `anti.*` pattern will do the trick.
Semantic-based methods are powerful tools to refine the taxonomy at the expense of a high dependency on the domain and the language. As a result, the automatization process is diminished.

4.3 Arbitrary domains

Note: This section is the printed version with three different executions (over arbitrary domains) of the interactive experiment presented as an IPython Notebook at `pykitools/doc/notebooks/Experiment - Arbitrary Exploration`.

4.3.1 Category:Political spectrum

Raw category graph

The first step is to generate the cetegory graph. The table 4.1 shows the initialization parameters as well as some other statistics about the category graph.

Root Node	Category:Political spectrum
Depth	4
Nodes	1338
- Categories	231
- Articles	1107
Edges	1523

Table 4.1: Statistics of the initial category graph.

Refinement steps

In the second step, the category graph is taken and refined by deleting categories with no articles and removing cycles. The table 4.2 shows the number of the empty categories found and cycles.

Empty categories	199
Cycles	0

Table 4.2: Statistics about refinement steps.

Multiparent conflict

In the third step, nodes with more than one parent are identified.

The table 4.3 shows some statistics about the refined graph and the identified conflicting nodes.

The figure 4.2 shows this graph. It is curious to see how political ideologies cluster themselves geographicaly according to their categorization in Left VS Right wing. It is also noticeable the need for a transitive reduction that will solve the correct taxonomization of articles like: `Peronism`, `Third Position` or `Centrism`.

Nodes	1137
- Categories	30
- Articles	1107
Edges	1304
Multiparent nodes	133

Table 4.3: Statistics after refinement steps and multiparent node identification.

To better see the problem, it is possible to reduce the category graph to a conflict graph that contains only the multiparent nodes and their path to the root node. The figure 4.3 shows the conflict reduction graph and the table 4.4 some statistics about this graph.

Nodes	152
- Categories	25
- Articles	127
Edges	319
Multiparent nodes	133

Table 4.4: Statistics of the reduced conflict graph.

Relevant terms

It is possible to see the most important terms used in the corpus of documents under the explored domain. This are the steps taken:

1. Extract the suitable features. A document is represented by the terms extracted from the readable text.

2. The documents represented as term vectors are used to create a Bag Of Words and then the terms are weighted with TF-IDF.

By averaging the term weigths it is possible to generate a word cloud diagram like de figure 4.1.
If we explore the terms it is possible to see that they are highly related to the politic domain (parti-, polit-, social-, group-, peopl-, nation, movement, member), but a further exploration reveals that there are also terms that are not so ideology agnostic and introduce a bias (communist, socialist, marxist soviet). This scenerario is generated because there are more left-wing articles than left-wing ones, see figure 4.2. It will be interesting to open a discussion among Wikipedia editors trying to compare this discovery against their Neutral Point of View Policy [52].

Figure 4.1: *Political spectrum* word cloud diagram.

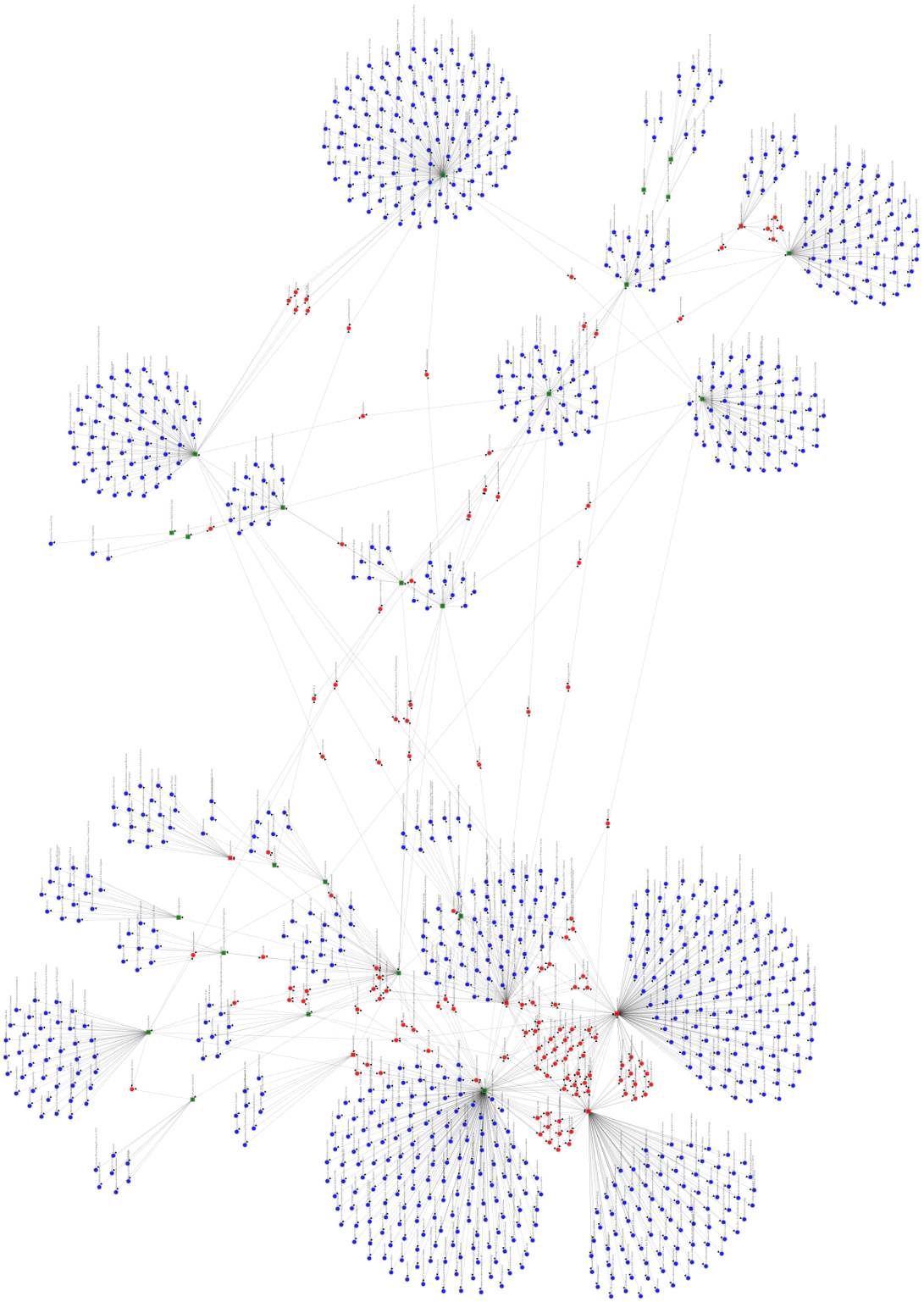

Figure 4.2: *Political spectrum* category graph (**green**: categories ,**blue**: articles and **red**: multiparent nodes).

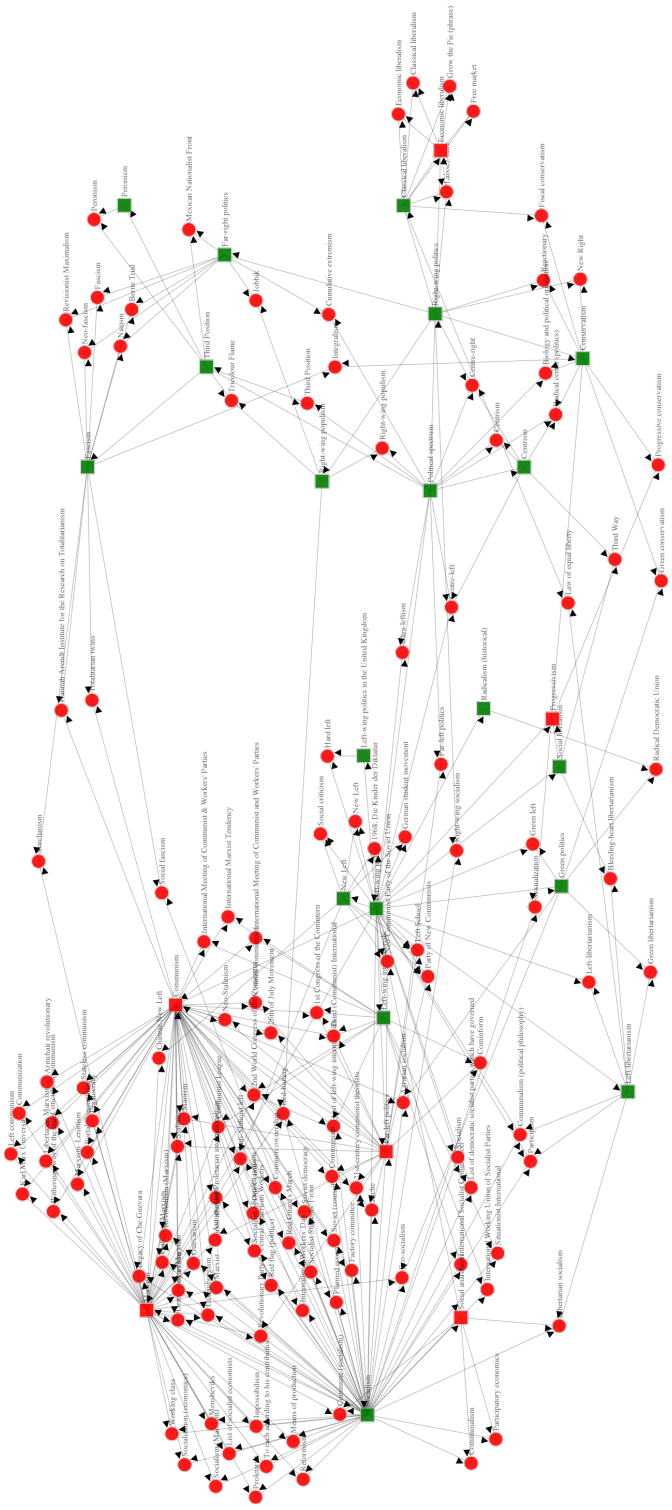

Figure 4.3: *Political spectrum* conflict graph.

4.3.2 Category:Religious faiths, traditions, and movements

Raw category graph

The first step is to generate the cetegory graph. The table 4.5 shows the initialization parameters as well as some other statistics about the category graph.

Root Node	Category:Religious faiths, traditions, and movements
Depth	4
Nodes	10674
- Categories	1676
- Articles	8998
Edges	12920

Table 4.5: Statistics of the initial category graph.

Refinement steps

In the second step, the category graph is taken and refined by deleting categories with no articles and removing cycles. The table 4.6 shows the number of the empty categories found and cycles.

Empty categories	1261
Cycles	0

Table 4.6: Statistics about refinement steps.

Multiparent conflict

In the third step, nodes with more than one parent are identified.

The table 4.7 shows some statistics about the refined graph and the identified conflicting nodes.

The figure 4.5 shows this graph. This graph is huge and the generation has been a real problem since web browsers ran out of memory. It is possible to distinguish several clusters, related to hinduism, buddhism and Abrahamic religions (Judaism, Christianism, Islam ...).

Nodes	9394
- Categories	396
- Articles	8998
Edges	11469
Multiparent nodes	1679

Table 4.7: Statistics after refinement steps and multiparent node identification.

To better see the problem, it is possible to reduce the category graph to a conflict graph that contains only the multiparent nodes and their path to the root node. The figure 4.6 shows the conflict reduction graph and the table 4.8 some statistics about this graph.

Nodes	1913
- Categories	345
- Articles	1568
Edges	3988
Multiparent nodes	1679

Table 4.8: Statistics of the reduced conflict graph.

Relevant terms

It is possible to see the most important terms used in the corpus of documents under the explored domain. This are the steps taken:

1. Extract the suitable features. A document is represented by the terms extracted from the readable text.

2. The documents represented as term vectors are used to create a Bag Of Words and then the terms are weighted with TF-IDF.

By averaging the term weigths it is possible to generate a word cloud diagram like de figure 4.4. The extracted terms makes sense and give a vision of what religion englobes, from strictly religious terms (religi-, god, templ-, deiti, church, soul), to others related to social life (school, tradit-, citi-, peopl, state, work) and even others with a negative meaning (repres, war, die).

Figure 4.4: *Religious faiths, traditions, and movements* word cloud diagram.

Figure 4.5: *Religious faiths, traditions, and movements* category graph (**green**: categories ,**blue**: articles and **red**: multiparent nodes).

4.3.3 Category:Sports by type

Raw category graph

The first step is to generate the cetegory graph. The table 4.9 shows the initialization parameters as well as some other statistics about the category graph.

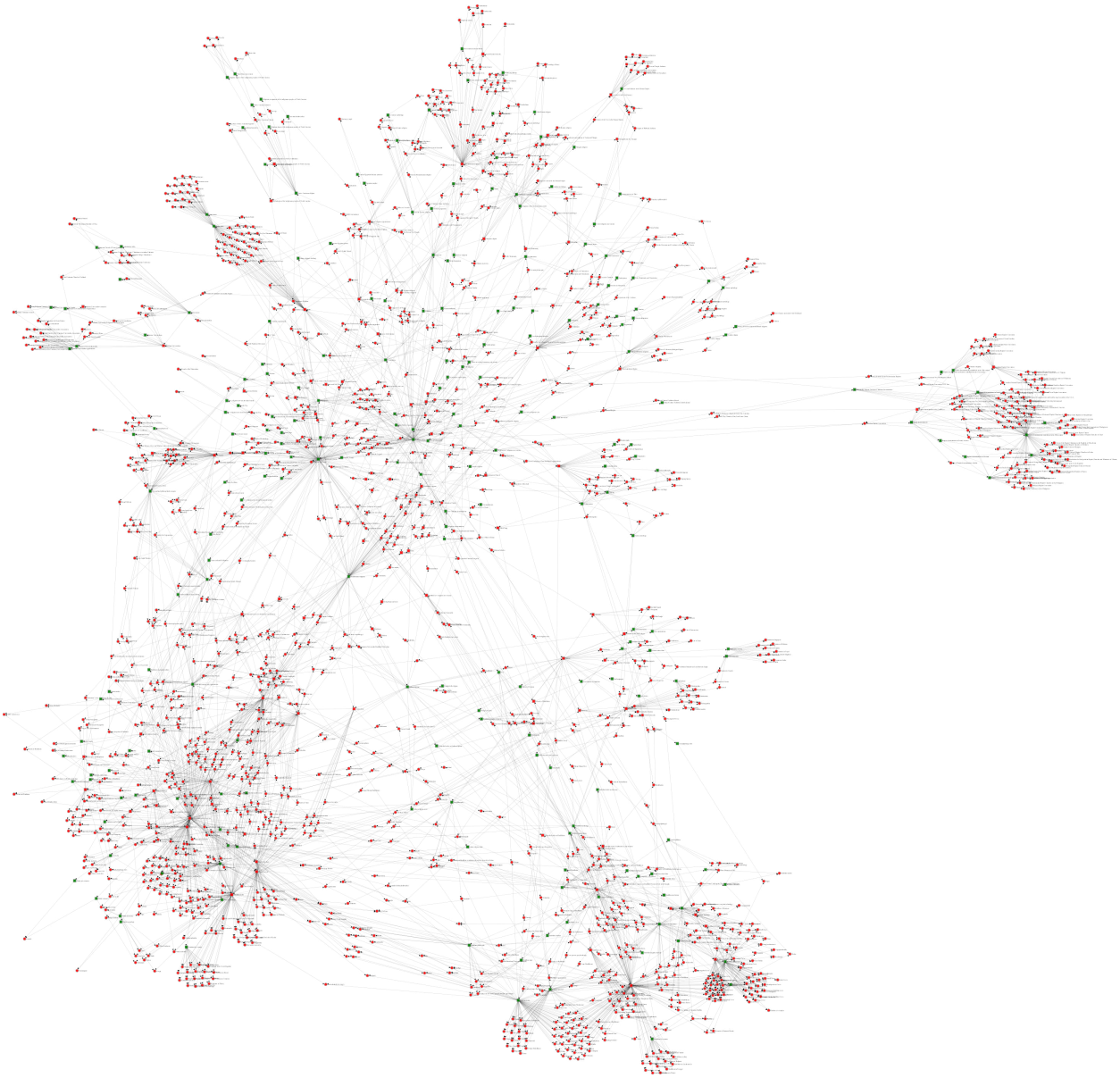

Figure 4.6: *Religious faiths, traditions, and movements* conflict graph.

Refinement steps

In the second step, the category graph is taken and refined by deleting categories with no articles and removing cycles. The table 4.10 shows the number of the empty categories found and cycles.

Root Node	Category:Sports by type
Depth	4
Nodes	9949
- Categories	3423
- Articles	6526
Edges	12099

Table 4.9: Statistics of the initial category graph.

Empty categories	2904
Cycles	0

Table 4.10: Statistics about refinement steps.

Multiparent conflict

In the third step, nodes with more than one parent are identified.

The table 4.11 shows some statistics about the refined graph and the identified conflicting nodes.

The figure 4.8 shows this graph. In this example, no particular clusters can be identified, the graph has a mesh topology.

Nodes	7024
- Categories	498
- Articles	6526
Edges	9000
Multiparent nodes	1313

Table 4.11: Statistics after refinement steps and multiparent node identification.

To better see the problem, it is possible to reduce the category graph to a conflict graph that contains only the multiparent nodes and their path to the root node. The figure 4.9 shows the conflict reduction graph and the table 4.12 some statistics about this graph. It can bee seen that there are a lot of coupled nodes, this is due to the multi-perspective nature of this taxonomy. A sport can be played with a ball, be a team sport and also be a water sport, so it will have at leas three parents. This is the hardest problem and the solution will be to particularize the root node so the taxonomy starts in a deeper level where fortunately appear less aspects.

Nodes	1494
- Categories	445
- Articles	1049
Edges	3470
Multiparent nodes	1313

Table 4.12: Statistics of the reduced conflict graph.

Relevant terms

It is possible to see the most important terms used in the corpus of documents under the explored domain. This are the steps taken:

1. Extract the suitable features. A document is represented by the terms extracted from the readable text.
2. The documents represented as term vectors are used to create a Bag Of Words and then the terms are weighted with TF-IDF.

By averaging the term weigths it is possible to generate a word cloud diagram like de figure 4.7. The terms found are related to de sport world like: game, sport, event, team, competit-...

Figure 4.7: *Sports by type* word cloud diagram.

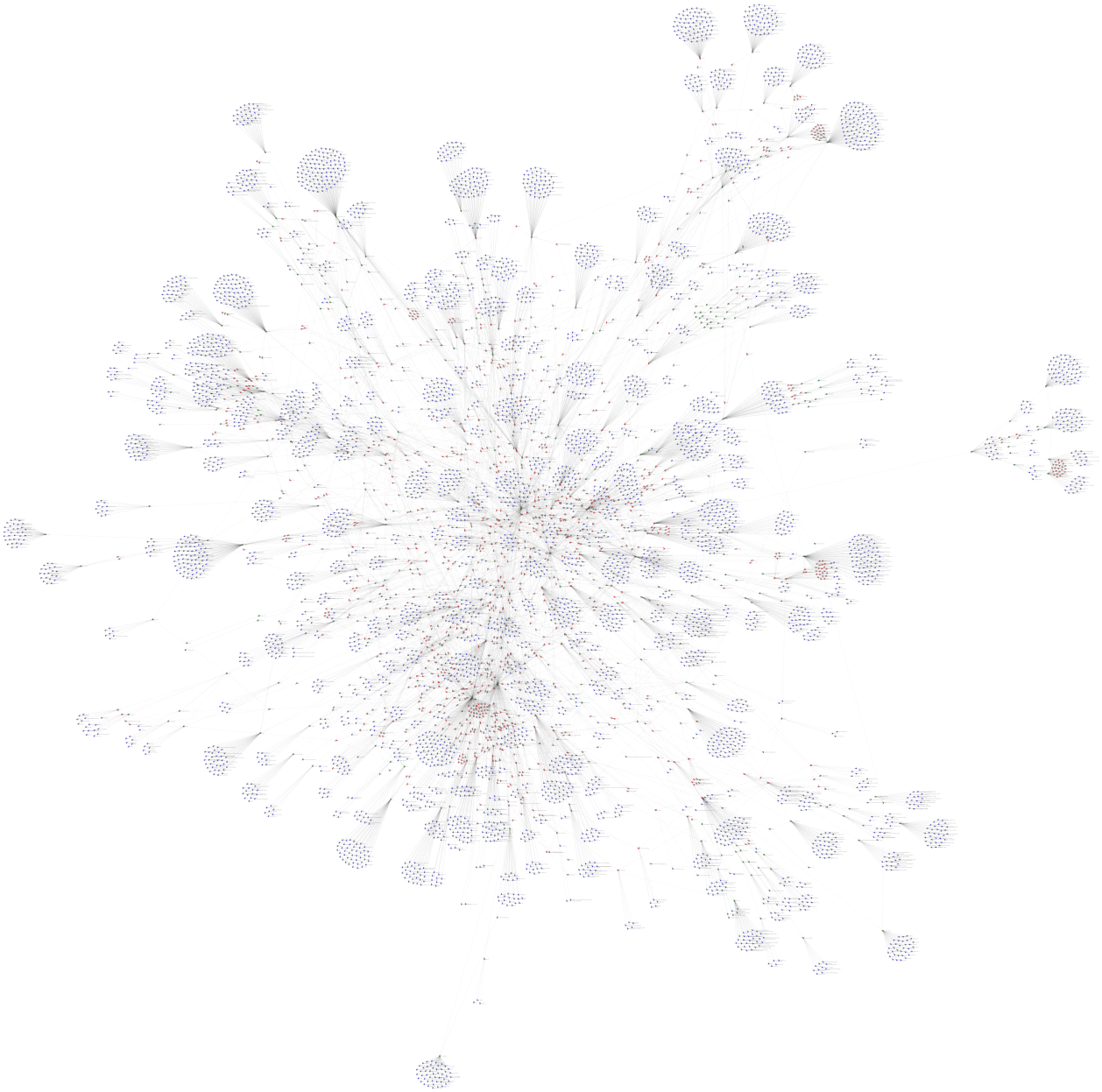

Figure 4.8: *Sports by type* category graph (**green**: categories ,**blue**: articles and **red**: multiparent nodes).

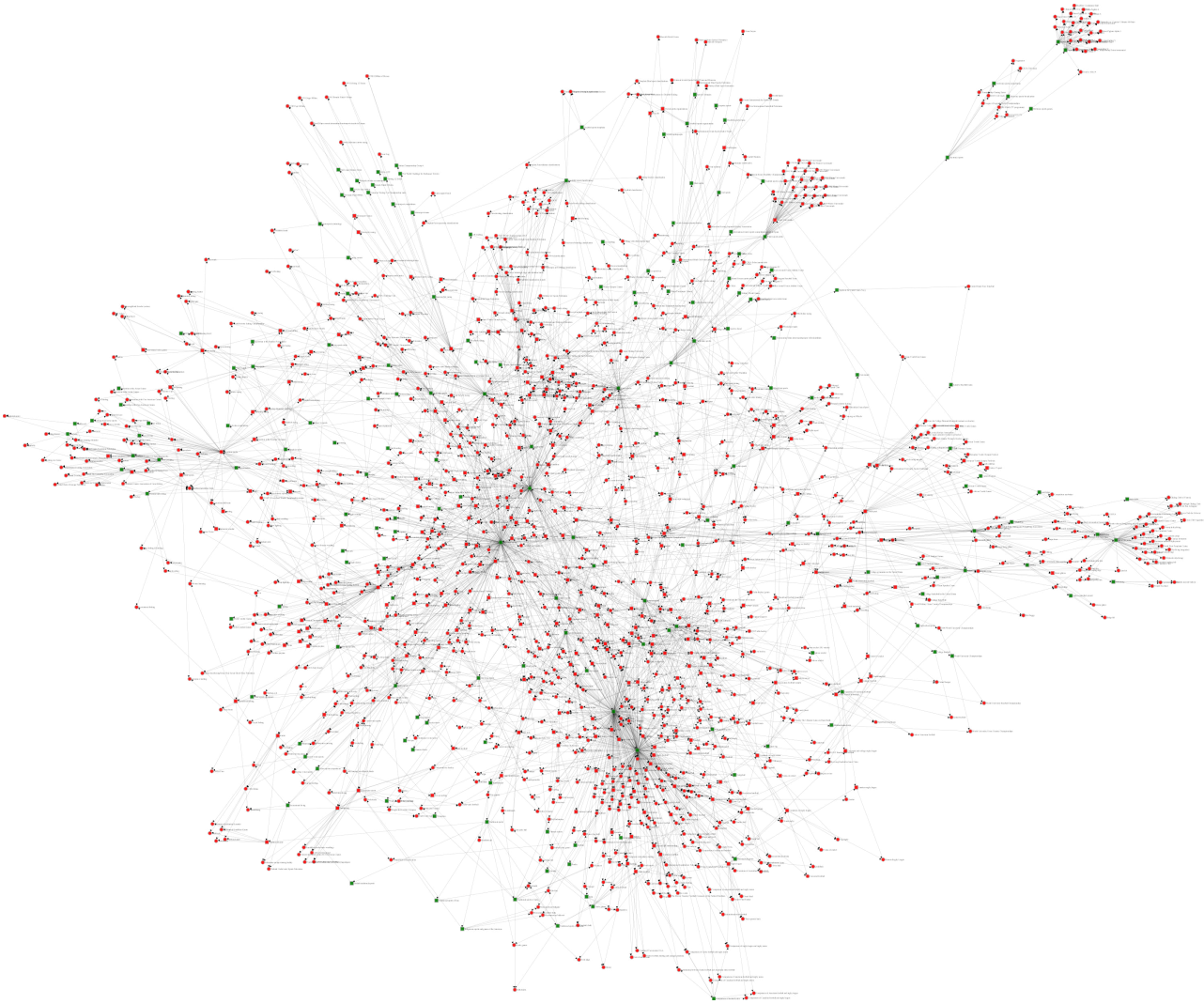

Figure 4.9: *Sports by type* conflict graph.

Chapter 5

Future work

The proposed architecture and some basic algorithms are implemented, nevertheless here are points where the framework can be improved.

- **Taxonomy layer**
  - Implement and improve more conflict detection/solving methods, specially semantic ones.
  - Investigate in more representative/efficient visualizations (i.e. it will be useful to colapse conflict-free subgraphs of a category graph).
- **Similarity layer**
  - Implement and test clustering methods.
  - Export and generate dendrogram views
  - Use dendrograms as a refinement step for the taxonomy induction.
  - Enrich taxonomies with information extracted from similarity models: TF-IDF, PLSI, LSI,etc.
- **Evaluation layer**
  - Implement an evaluation layer.
  - Apart from the SOA methods, it could be useful some sampling method to test the variance in a method.
- **Documentation**
  - Generate more experiments for the IPython Notebooks.

This are the possible working lines to carry on with this project.

- **Commercial exploitation**
  It should not be hard to adapt the project to create a Software as a Service (SaaS) application that generates taxonomies over demand. It could also be possible to generate text classifiers or summarizers over demand. The category graph viewer could also be useful to see other graph types and improve the human data exploration process.

- **Investigation lines**
  The multiparent node conflict is far away to be fully solved and more investigation should be done work it out. As seen in 4.2.1 there is also a need to mix complete-automatic techniques such as the transitive reduction with semantic ones to obtain satisfactory results. Work should be done in making semantic methods less domain dependant (¿maybe with more taxonomies?). There also a huge margin to explore in ontology evaluation. It will be very interesting to mix taxonomy evaluation techniques (ontology alignment, structural measures) with clustering ones (purity and information teoretical) in order to get new metrics and compare a category graph wit a dendrogram.

Chapter 6

Conclusion

This thesis makes an extensive review of the state of the art on taxonomy induction techniques as well as ontology evaluation methods. It claims the need for fast and arbitrary-domain taxonomy generation and justifies the use of the Wikipedia encyclopedia as the chosen dataset. A framework to explore and generate taxonomies is proposed and implemented. In the experiments chapter, two statements are successfully refuted: the Wikipedia categorization system forms an acyclic directed graph, and the longest path between two nodes is equivalent to the taxonomic organization. Finally the framework is tested with three arbitrary domains to see its exploratory power.

Bibliography

[1] Aho, A. V., Garey, M. R., and Ullman, J. D. The transitive reduction of a directed graph. *SIAM Journal on Computing 1*, 2 (1972), 131–137.

[2] Aitchison, J., Gilchrist, A., and Bawden, D. *Thesaurus construction and use: a practical manual*. Psychology Press, 2000.

[3] Bostock, M. D3.js offical page, 2014. http://d3js.org/ (Accessed 2014-09).

[4] Brank, J., Grobelnik, M., and Mladenić, D. A survey of ontology evaluation techniques. In *In In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005* (2005).

[5] Brewster, C., Alani, H., and Dasmahapatra, A. Data driven ontology evaluation. In *In Int. Conf. on Language Resources and Evaluation* (2004).

[6] Cambridge University Press. Definition of taxonomy from the cambridge advanced learners dictionary and thesaurus, 2014. http://dictionary.cambridge.org/dictionary/british/taxonomy (Accessed 2014-09).

[7] Capocci, A., Rao, F., and Caldarelli, G. Taxonomy and clustering in collaborative systems: The case of the on-line encyclopedia wikipedia. *EPL (Europhysics Letters) 81*, 2 (2008), 28006.

[8] Cimiano, P., Hotho, A., and Staab, S. Clustering concept hierarchies from text. In *Proceedings of the Conference on Lexical Resources and Evaluation (LREC)* (2004).

[9] Cycorp. Opencyc, 2014. http://www.cyc.com/platform/opencyc (Accessed 2014-09).

[10] Domingos, P. A Few Useful Things to Know About Machine Learning. *Communications of the ACM 55*, 10 (OCT 2012), 78–87. PT: J; NR: 24; TC: 8; J9: COMMUN ACM; PG: 10; GA: 012DN; UT: WOS:000309215800024.

[11] Giles, J. Internet encyclopaedias go head to head. *Nature 438*, 7070 (2005), 900–901.

[12] GitHub, . Github official page, 2014. https://github.com/ (Accessed 2014-09).

[13] Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), ACM, pp. 50–57.

[14] Hovy, E. Comparing sets of semantic relations in ontologies. In *The semantics of relationships*. Springer, 2002, pp. 91–110.

[15] Hovy, E., Kozareva, Z., and Riloff, E. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2* (Stroudsburg, PA, USA, 2009), EMNLP '09, Association for Computational Linguistics, pp. 948–957.

[16] Hu, X., Zhang, X., Lu, C., Park, E. K., and Zhou, X. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2009), KDD '09, ACM, pp. 389–396.

[17] IPython Development Team. Ipython notebook official page, 2014. http://ipython.org/notebook.html (Accessed 2014-09).

[18] Jain, A., Murty, M., and Flynn, P. Data clustering: A review. *Acm Computing Surveys 31*, 3 (SEP 1999), 264–323. PT: J; NR: 204; TC: 2645; J9: ACM COMPUT SURV; PG: 60; GA: 302KZ; UT: WOS:000086365400002.

[19] Jiang, J. J., and Conrath, D. W. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR cmp-lg/9709008* (1997).

[20] Kozareva, Z., and Hovy, E. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2010), EMNLP '10, Association for Computational Linguistics, pp. 1110–1118.

[21] Krishnaprasad, S. Uses and abuses of amdahl's law. *Journal of Computing Sciences in Colleges 17*, 2 (2001), 288–293.

[22] Liu, T., Liu, S., Chen, Z., and Ma, W.-Y. An evaluation on feature selection for text clustering. In *ICML* (2003), vol. 3, pp. 488–495.

[23] Maedche, A., and Staab, S. Measuring similarity between ontologies. In *Knowledge engineering and knowledge management: Ontologies and the semantic web*. Springer, 2002, pp. 251–263.

[24] Mediawiki Foundation. Markup spec, 2014. http://www.mediawiki.org/wiki/Markup_spec (Accessed 2014-09).

[25] Miller, G. A. Wordnet: A lexical database for english. *Commun. ACM 38*, 11 (Nov. 1995), 39–41.

[26] NetworkX Developer Team. Networkx, 2014. https://networkx.github.io/ (Accessed 2014-09).

[27] New Idea Engineering, . What's the difference between taxonomies and ontologies?, 2009. http://www.ideaeng.com/taxonomies-ontologies-0602 (Accessed 2014-09).

[28] Oxford University Press. Definition of taxonomy in oxford dictionary, 2014. http://www.oxforddictionaries.com/definition/english/taxonomy (Accessed 2014-09).

[29] Page, L., Brin, S., Motwani, R., and Winograd, T. The pagerank citation ranking: Bringing order to the web. 1999.

[30] Petersen, W. A set-theoretical approach for the induction of inheritance hierarchies. *Electronic Notes in Theoretical Computer Science 53* (2004), 296–308.

[31] Pidcock, W. What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model?, 2010. http://infogrid.org/trac/wiki/Reference/PidcockArticle (Accessed 2014-09).

[32] Ponzetto, S. P., and Strube, M. Deriving a large scale taxonomy from wikipedia. In *AAAI* (2007), vol. 7, pp. 1440–1445.

[33] Ponzetto, S. P., and Strube, M. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence 175*, 9–10 (2011), 1737 – 1756.

[34] Princeton University. Wordnet, 2014. http://wordnet.princeton.edu/ (Accessed 2014-09).

[35] Priss, U. Linguistic applications of formal concept analysis. In *Formal Concept Analysis*. Springer, 2005, pp. 149–160.

[36] Python Software Foundation. Python official page, 2014. https://www.python.org/ (Accessed 2014-09).

[37] Rosenblatt, G. What is network density – and how do you calculate it?, 2013. http://www.the-vital-edge.com/what-is-network-density/ (Accessed 2014-09).

[38] Ruiz-Casado, M., Alfonseca, E., and Castells, P. Automatic assignment of wikipedia encyclopedic entries to wordnet synsets. In *Advances in Web Intelligence*, P. Szczepaniak, J. Kacprzyk, and A. Niewiadomski, Eds., vol. 3528 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 380–386.

[39] Salton, G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[40] Salton, G., and Yang, C.-S. On the specification of term values in automatic indexing. *Journal of documentation 29*, 4 (1973), 351–372.

[41] Stevenson, M. Combining disambiguation techniques to enrich an ontology. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence (ECAI-02) workshop on" Machine Learning and Natural Language Processing for Ontology Engineering", Lyon, France* (2002).

[42] Taxonomy Strategies. Taxonomy glossary, 2006. http://taxonomystrategies.com/html/Taxonomy_Glossary.pdf (Accessed 2014-09).

[43] Thai Open Source Software Center Ltd. The expat xml parser, 2007. http://expat.sourceforge.net/ (Accessed 2014-09).

[44] Thornton, K., and McDonald, D. W. Tagging wikipedia: Collaboratively creating a category system. In *Proceedings of the 17th ACM International Conference on Supporting Group Work* (New York, NY, USA, 2012), GROUP '12, ACM, pp. 219–228.

[45] U.S. National Library of Medicine. Mesh, 2014. http://www.nlm.nih.gov/mesh/ (Accessed 2014-09).

[46] Wikipedia. Definition of taxonomy from the wikipedia, 2014. http://en.wikipedia.org/wiki/Taxonomy_%28general%29 (Accessed 2014-09).

[47] Wikipedia. Definition of wikipedia from the wikipedia, 2014. http://en.wikipedia.org/wiki/Wikipedia (Accessed 2014-09).

[48] Wikipedia. Simple api for xml, 2014. http://en.wikipedia.org/wiki/Simple_API_for_XML (Accessed 2014-09).

[49] Wikipedia. Wikipedia:api, 2014. http://www.mediawiki.org/wiki/API:Main_page (Accessed 2014-09).

[50] Wikipedia. Wikipedia:categorization, 2014. http://en.wikipedia.org/wiki/Wikipedia:Categorization (Accessed 2014-09).

[51] Wikipedia. Wikipedia:copyrights, 2014. http://en.wikipedia.org/wiki/Wikipedia:Copyrights (Accessed 2014-09).

[52] Wikipedia. Wikipedia:neutral point of view, 2014. http://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view (Accessed 2014-09).

[53] Wikipedia. Wikipedia:size of wikipedia, 2014. http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia (Accessed 2014-09).

[54] Zhang, W., Yoshida, T., and Tang, X. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications 38*, 3 (2011), 2758–2765.